ATTACK DEFENSE

by PentesterAcademy

Name	WMI: Configure via Windows GUI
URL	https://attackdefense.com/challengedetails?cid=2075
Type	Services Exploitation: WMI

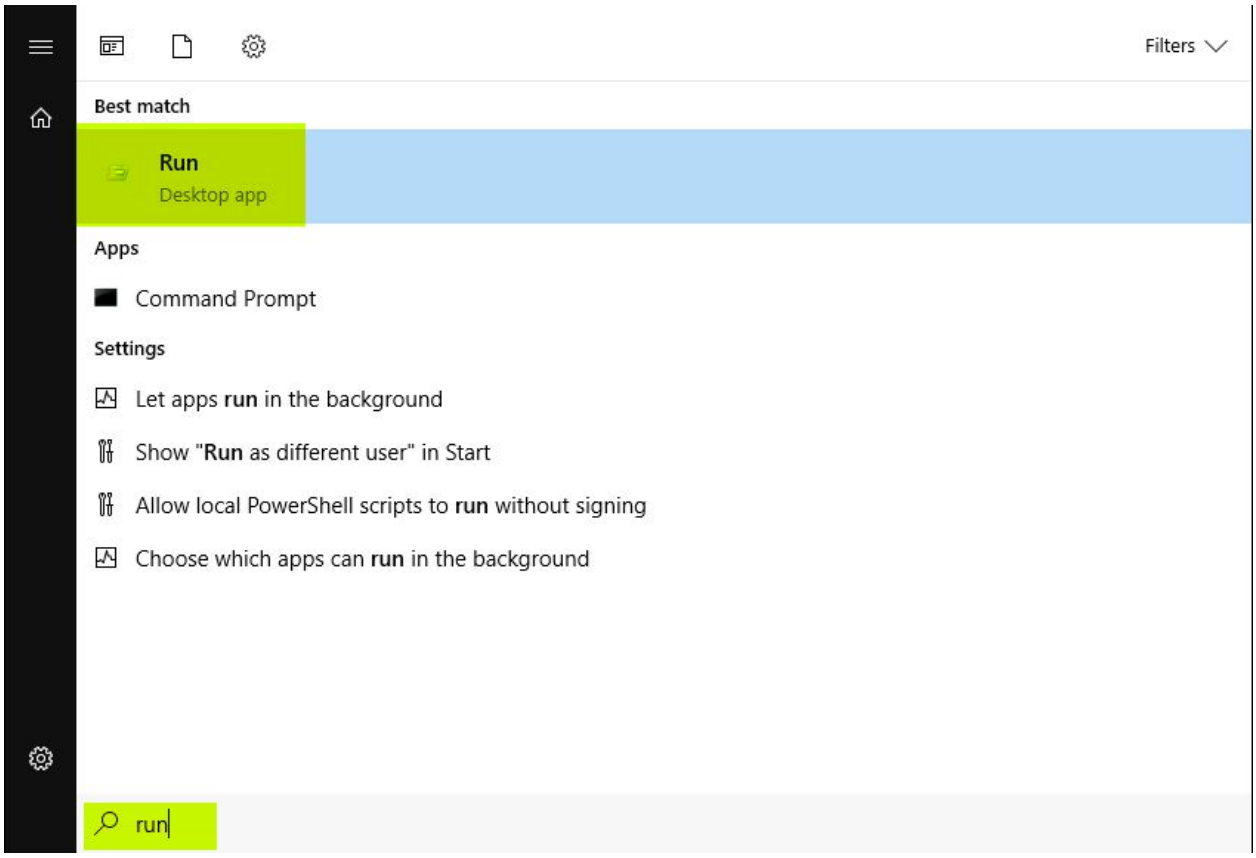
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Note: By default, if you are using Windows Server then, the WMI service is already up and running. You need to configure the service in order to access it remotely. In this manual, we are demonstrating how to configure WMI service and making necessary changes for learning purposes.

Configuration of WMI

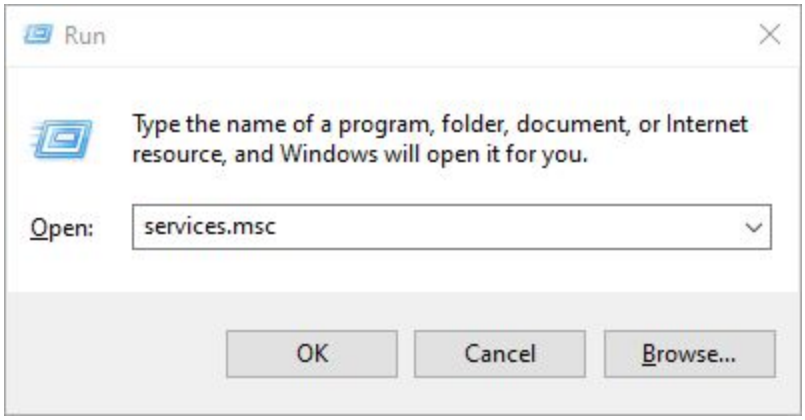
Note: Follow all the below steps on the "**Target Machine**" Also, The '**Windows + R**' would work on Linux systems. If you are using a Windows system, then it would conflict with your machine, and hence you won't be able to get the run prompt. You could manually open '**run**' by following the below steps:

1. Go to the Windows Start menu
2. Search for Run
3. Click on the Run App icon

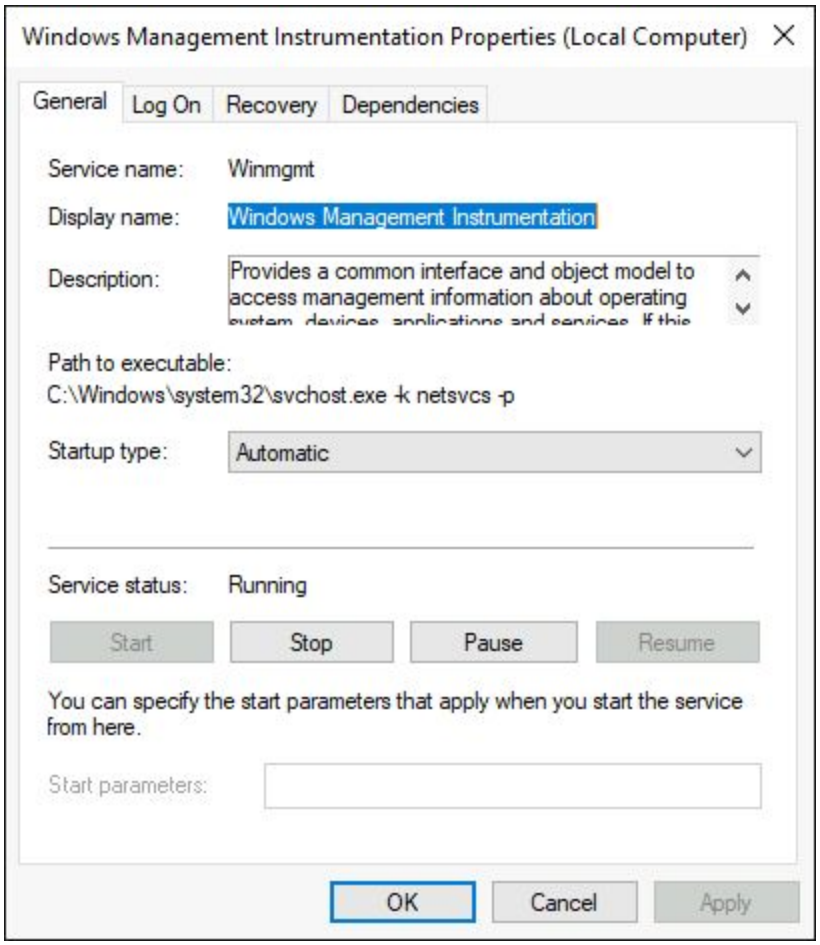


Step 1: Checking for wmi service status, if it's running or not. Open windows services manager.

Command: Press Windows + R, type services.msc in Run dialog, and hit the Enter key to open it.



Step 2: Locate the "**Windows Management Instrumentation**" i.e WMI service and check the status of the service.

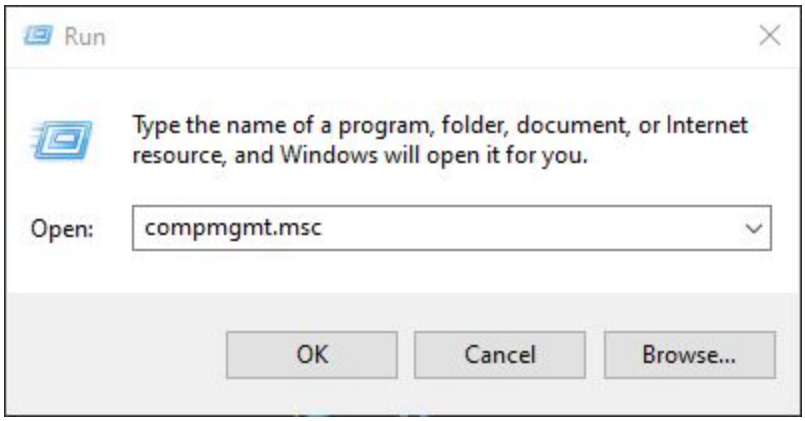


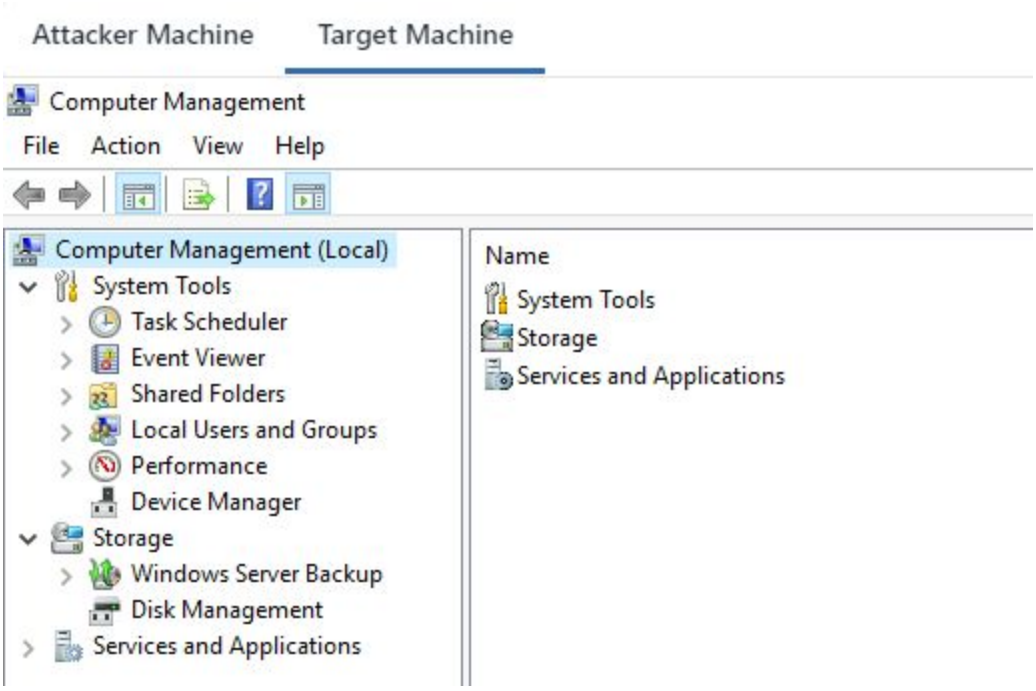
The WMI service is running

Creating a Demo User

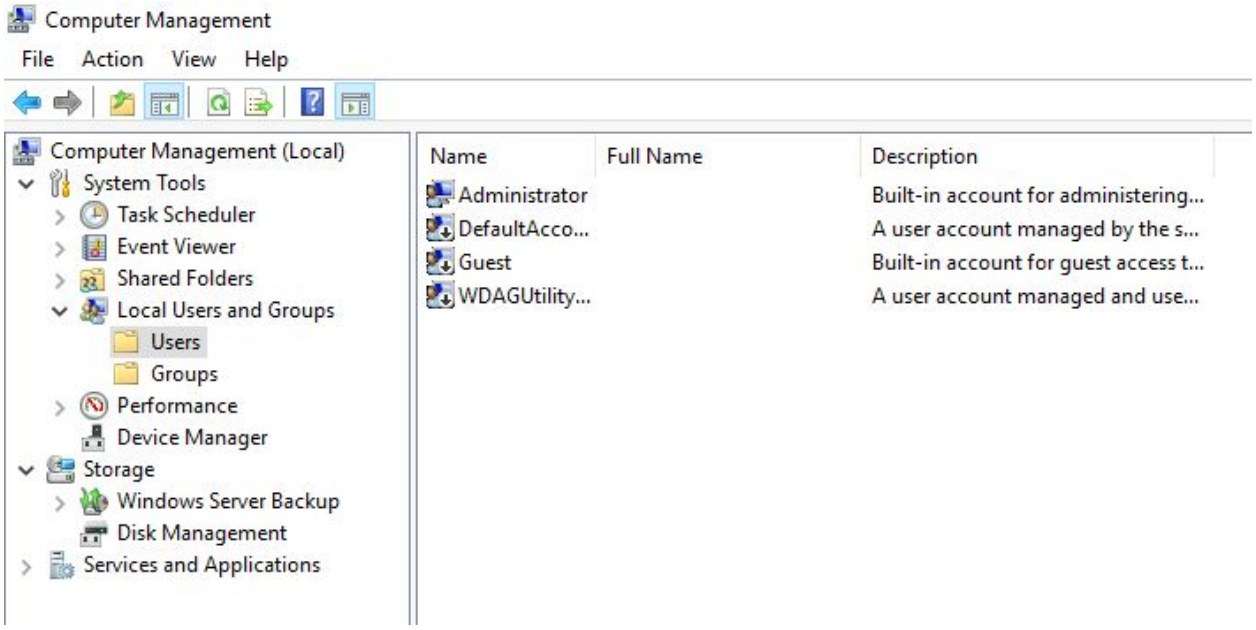
Step 1: We will create a demo user on the remote machine i.e **Target Machine** and the same user credentials we will use to execute commands on the remote machine from the client. i.e **Attacker Machine**

Press Windows + R, type **compmgmt.msc** in the Run dialog, and hit the Enter key to open it.

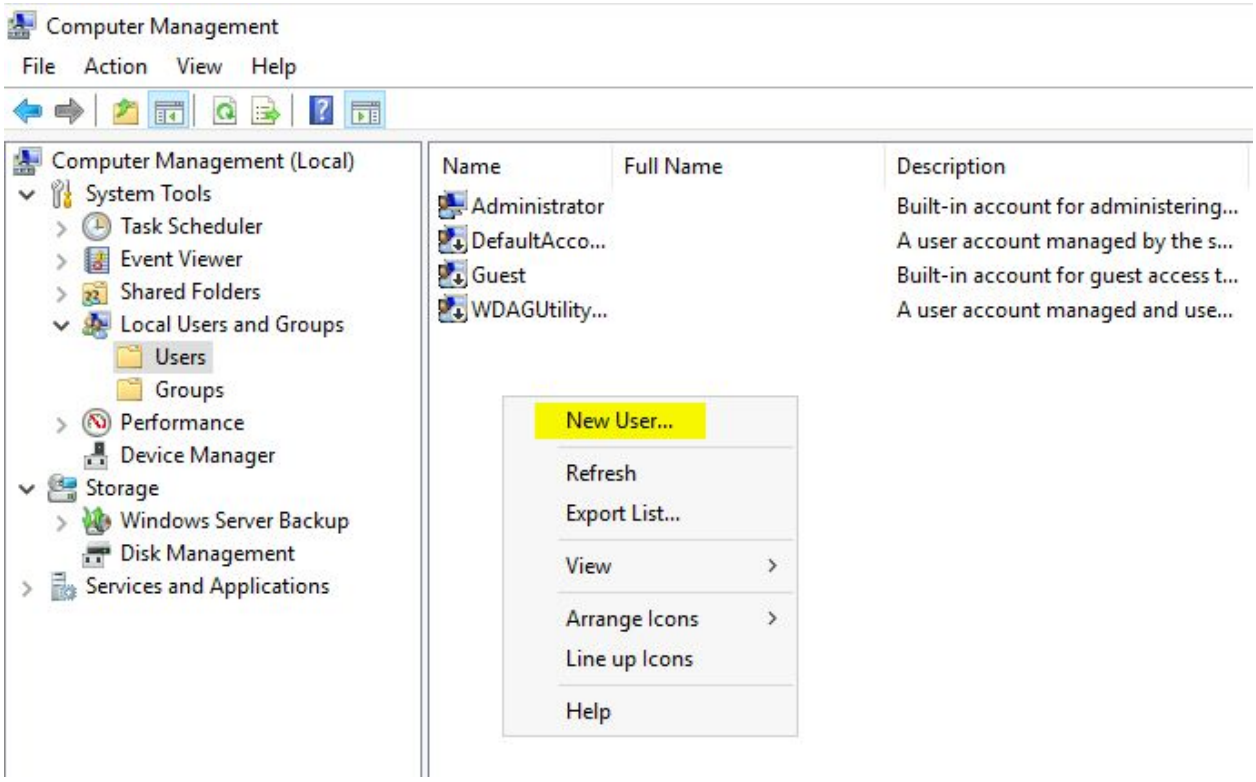


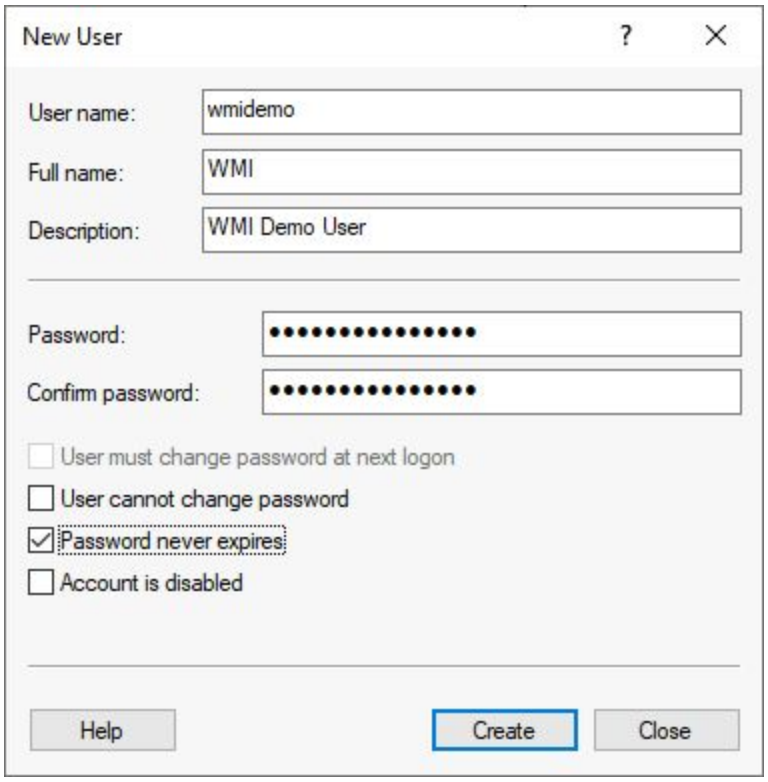


Step 2: Expand the Menu tree as follows: **Local Users and Groups > Users**

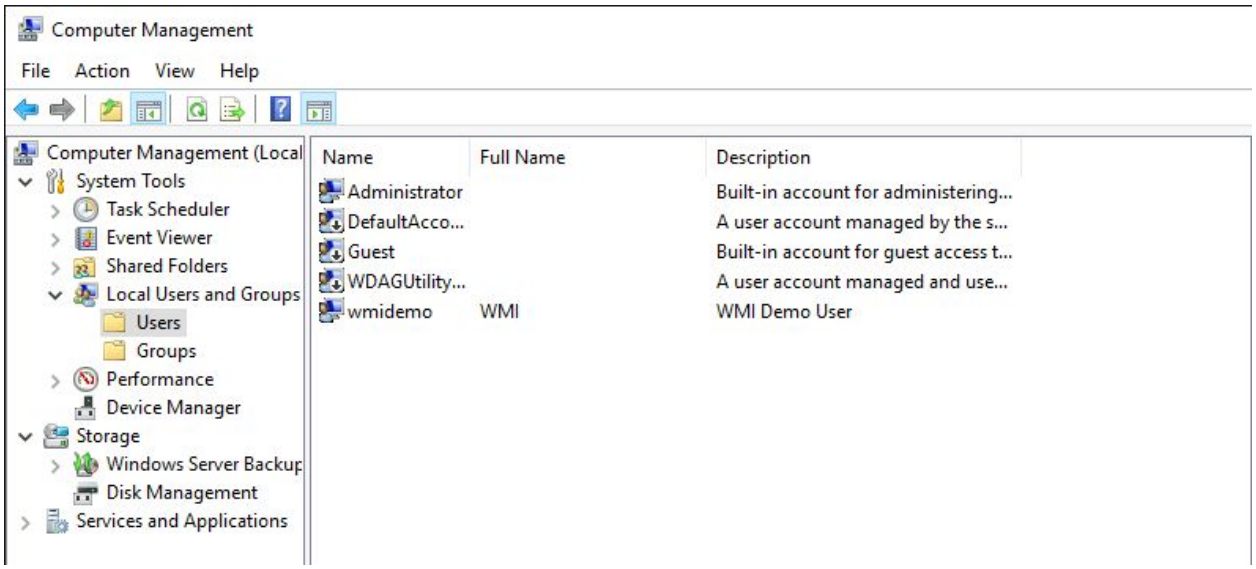


Step 3: We will create a demo user i.e: **wmidemo**:**password_123321** on the remote machine and the same user credentials we will use to execute commands.


Computer Management
File
Action
View
Help
Computer Management (Local)
System Tools
Task Scheduler
Event Viewer
Shared Folders
Local Users and Groups
Users
Groups
Performance
Device Manager
Storage
Windows Server Backup
Disk Management
Services and Applications
Name
Full Name
Description
Administrator
Built-in account for administering...
DefaultAcco...
A user account managed by the s...
Guest
Built-in account for guest access t...
WDAGUtility...
A user account managed and use...
New User...
Refresh
Export List...
View
Arrange Icons
Line up Icons
Help



After filling in the user details, click on “**Create**”



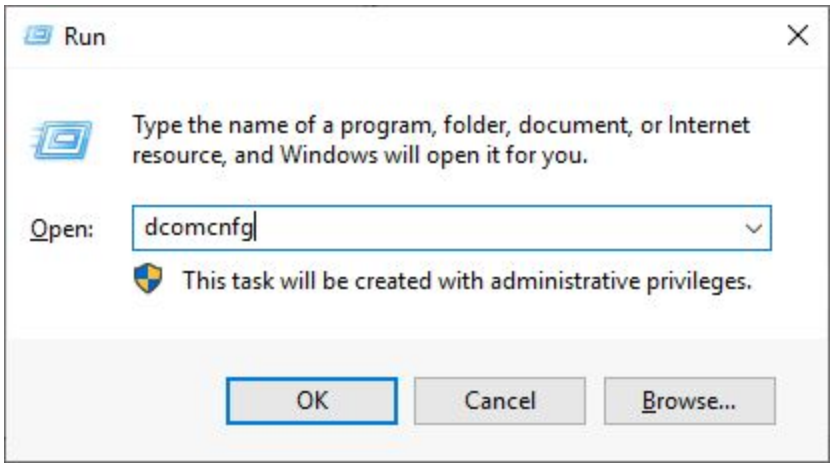
We have successfully created a user i.e "**wmidemo**".

By default, we cannot execute remote WMI queries using the created user. We need to give permission and allow the user to execute remote queries.

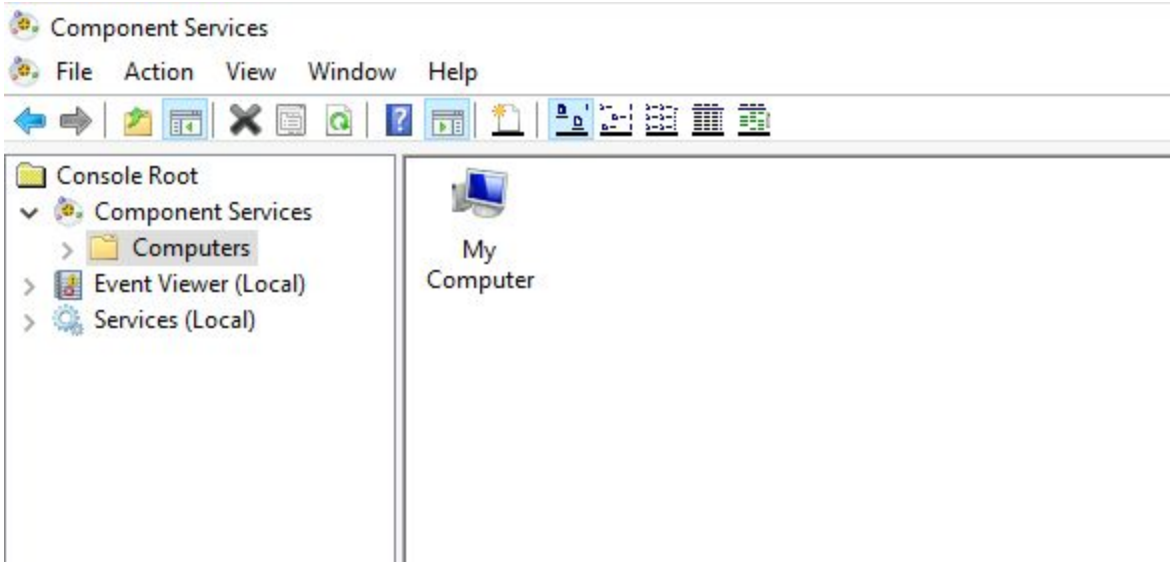
Granting WMI Access Permissions

Step 1: Granting WMI remote access permissions to wmidemo user.

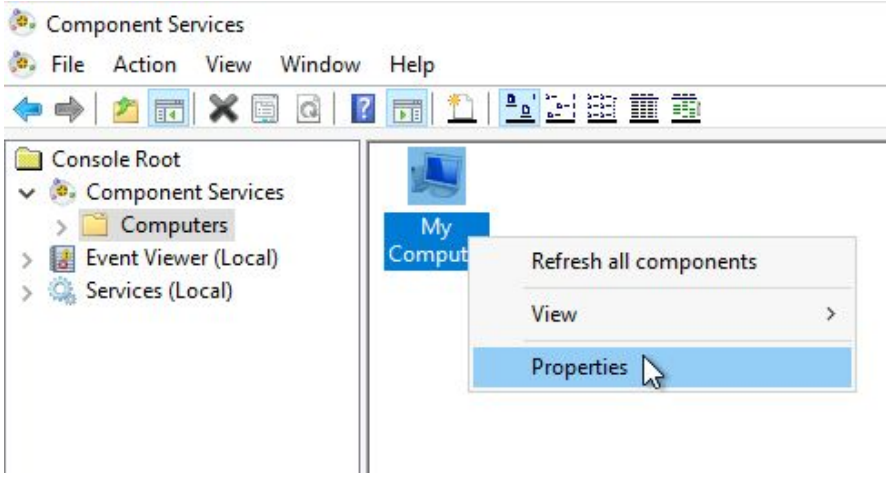
Press Windows + R, type **dcomcnfg** in Run dialog, and hit the Enter key to open it.

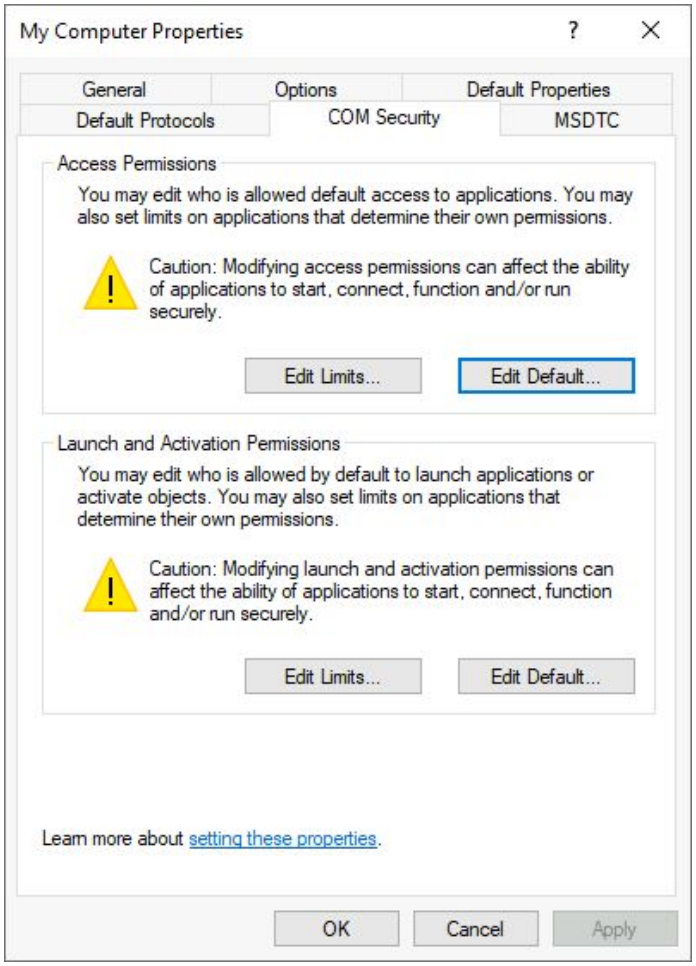


Step 2: Expand the Menu tree as follows: **Component Services > Computers > My Computer**



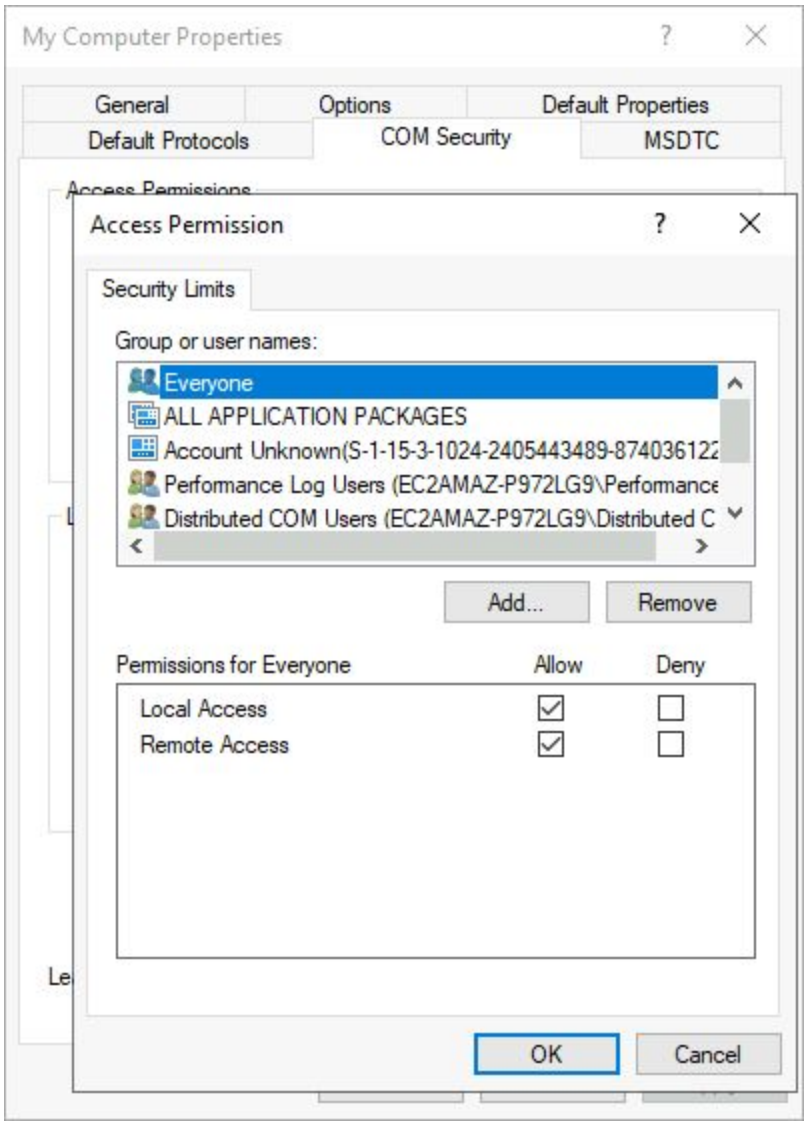
Right-click on "**My Computer**" and click on **'Properties**" then Navigate "**COM Security**"


Component Services
File Action View Window Help
Console Root
Component Services
Computers
Event Viewer (Local)
Services (Local)
My Comput
Refresh all components
View
Properties


My Computer Properties
General
Options
Default Properties
Default Protocols
COM Security
MSDTC
Access Permissions
You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.
Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.
Edit Limits...
Edit Default...
Launch and Activation Permissions
You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.
Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.
Edit Limits...
Edit Default...
Learn more about setting these properties.
OK
Cancel
Apply

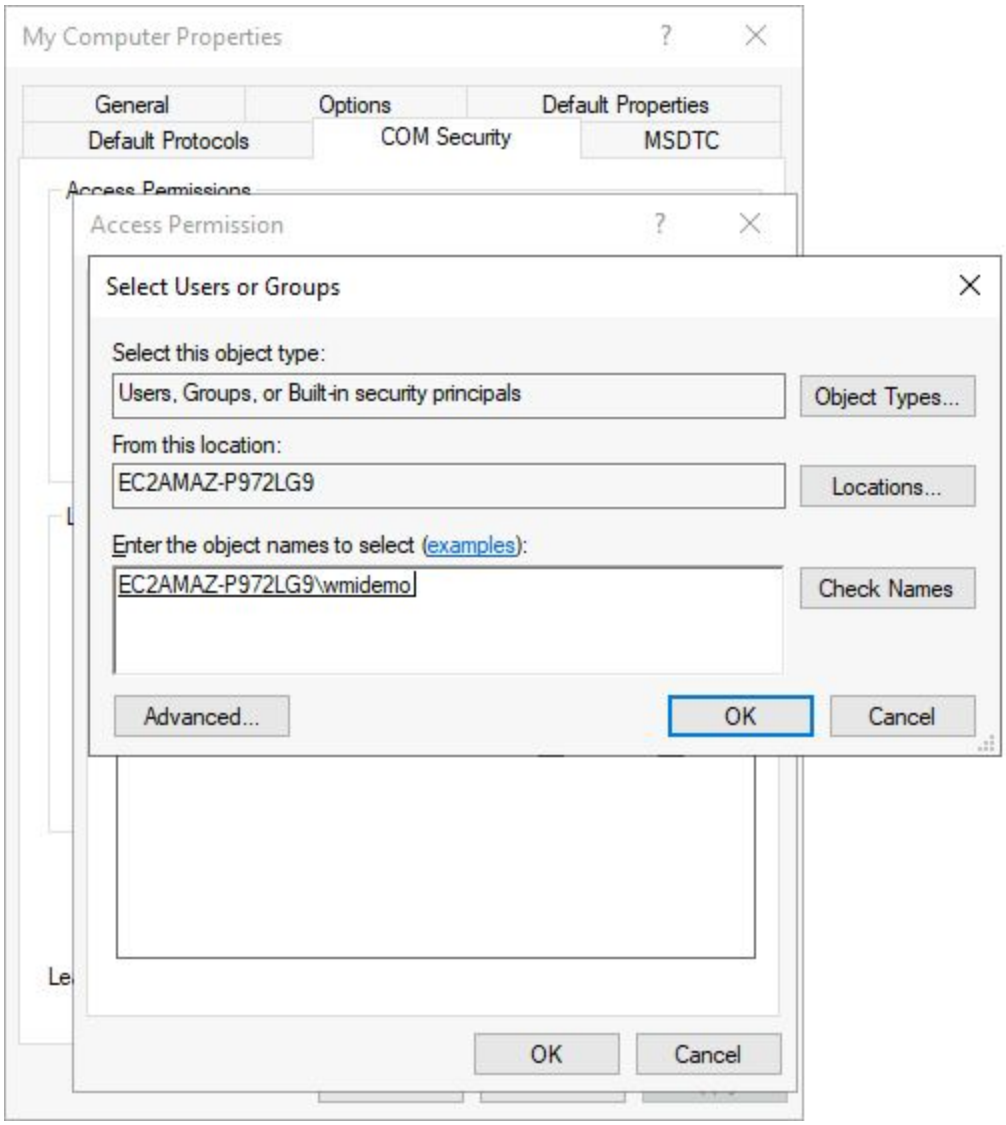
Step 3: We will give the permission to wmidemo user i.e "**Access Permissions**" and "Launch and Activation Permissions"

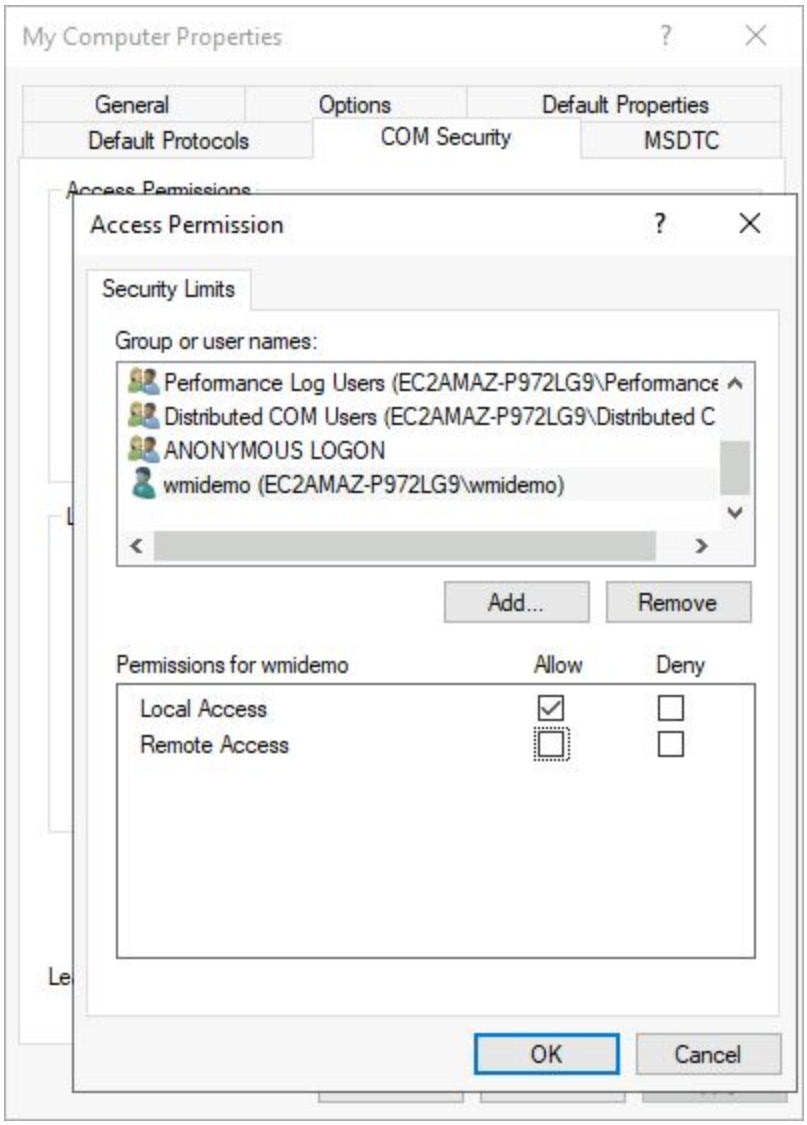
Click on "**Access Permissions: Edit Limits..**"



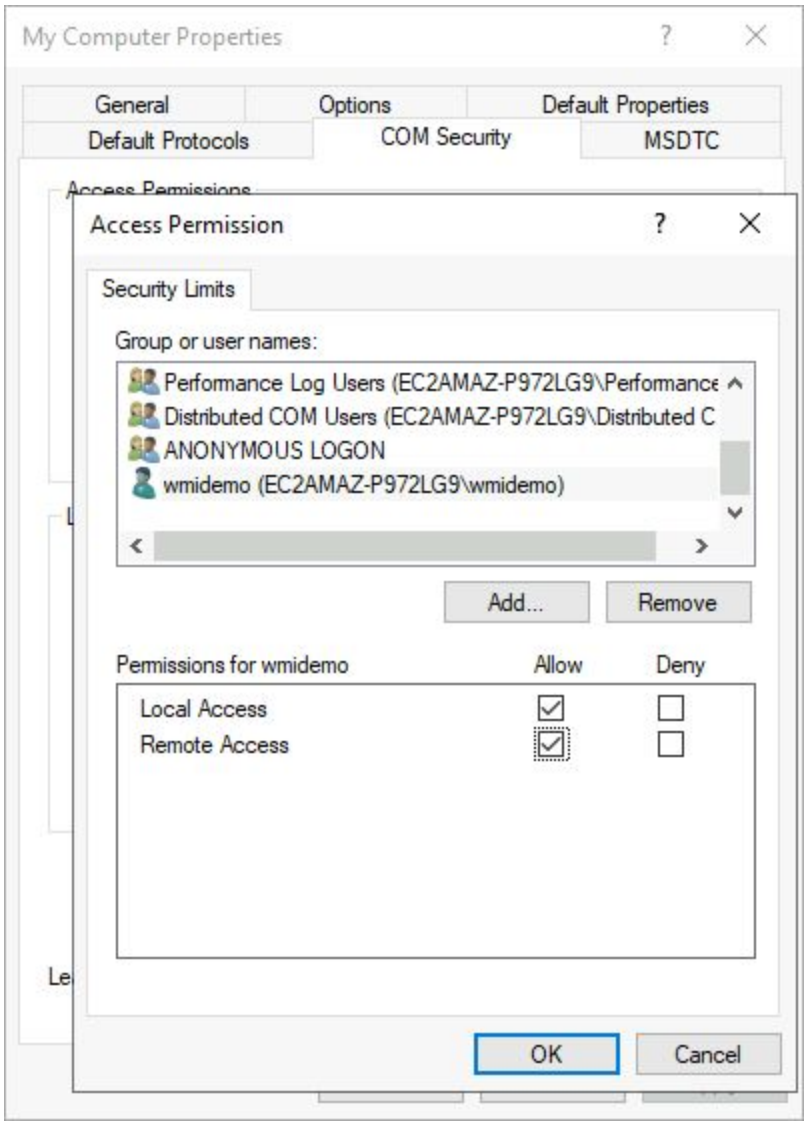
Note: By default "**Everyone**" has the "**Access Permissions**"

Click on "**Add..**" and type wmidemo user and click on "**Check Names**"


My Computer Properties
General
Options
Default Properties
Default Protocols
COM Security
MSDTC
Access Permissions
Access Permission
Select Users or Groups
Select this object type:
Users, Groups, or Built-in security principals
Object Types...
From this location:
EC2AMAZ-P972LG9
Locations...
Enter the object names to select (examples):
EC2AMAZ-P972LG9\wmidemo
Check Names
Advanced...
OK
Cancel
OK
Cancel

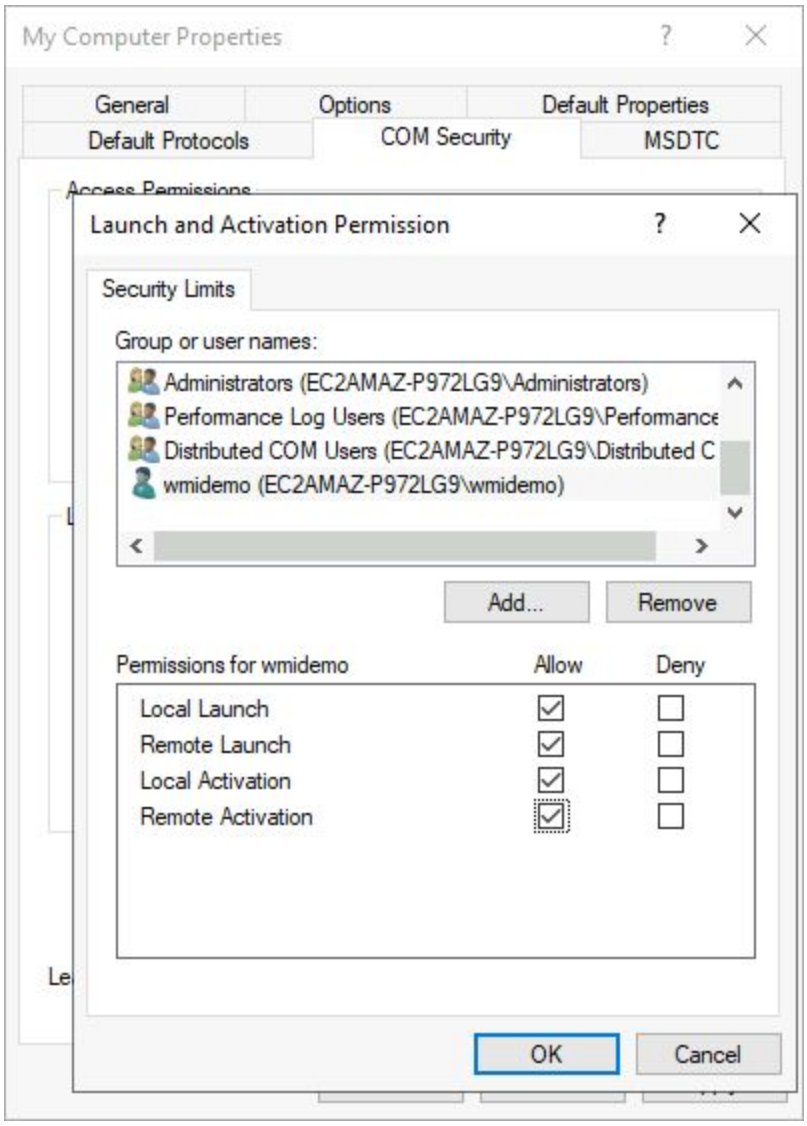


Currently, the user has only local access, check the remote access box to "**Allow**".



Click "**Ok**"

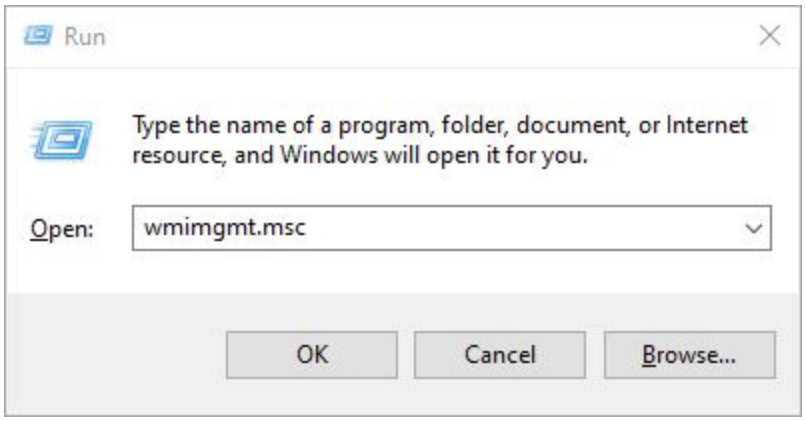
Click on the "**Launch and Activation Permissions**" and follow the exact same above steps and add **wmidemo** users with all the permissions.



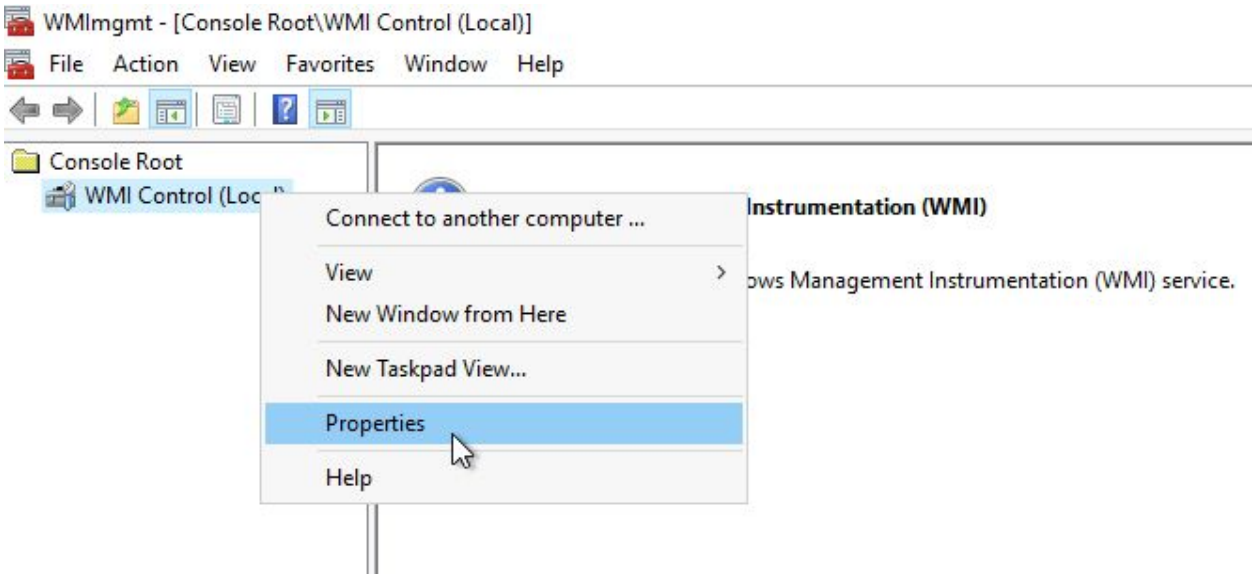
Click "**Apply**". We have successfully given WMI access permissions to wmidemo user.

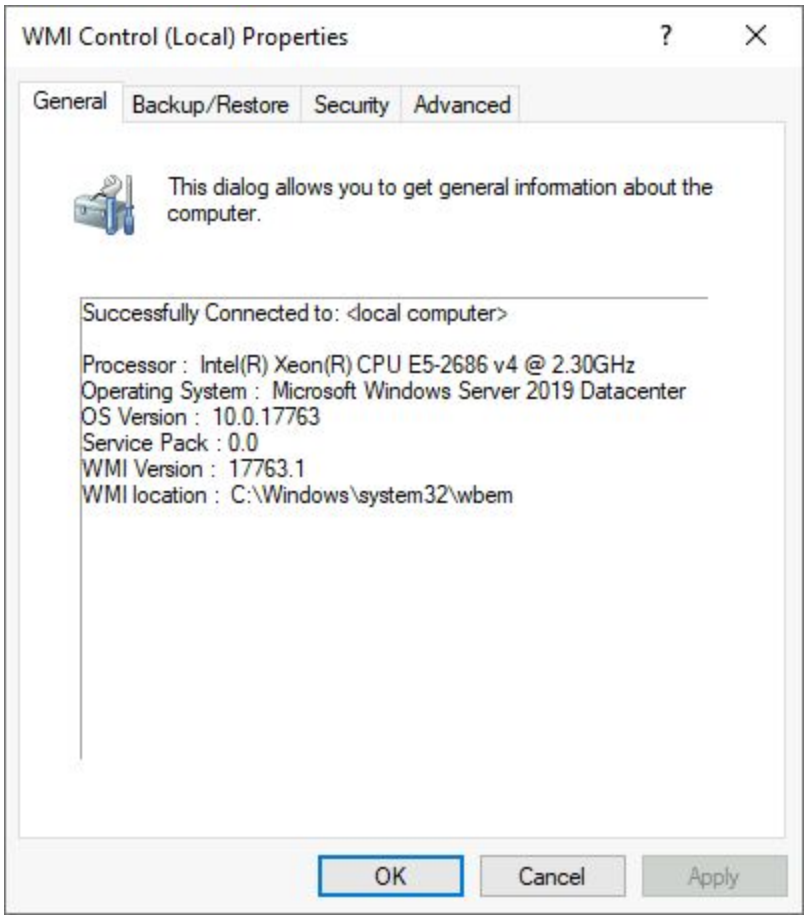
Step 4: We also need to give permission to "**wmidemo**" user from WMI Access by adding the user.

Press Windows + R, type **wmimgmt.msc** in the Run dialog, and hit the Enter key to open it.

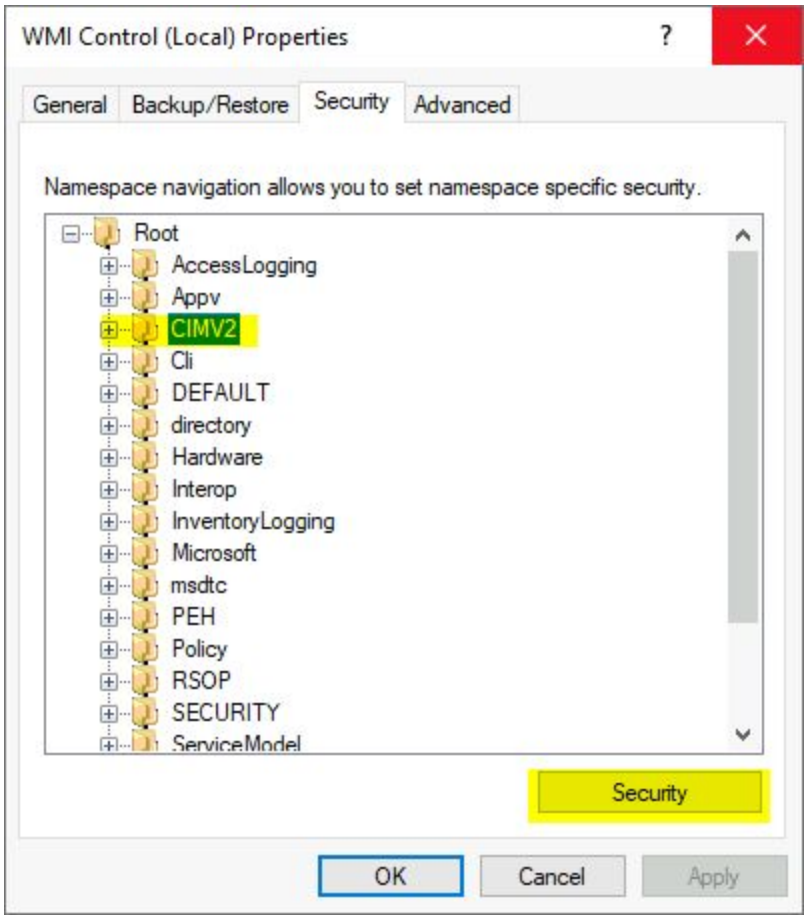


Step 5: Right-click on “WMI Control (Local)” and click on “**Properties**”

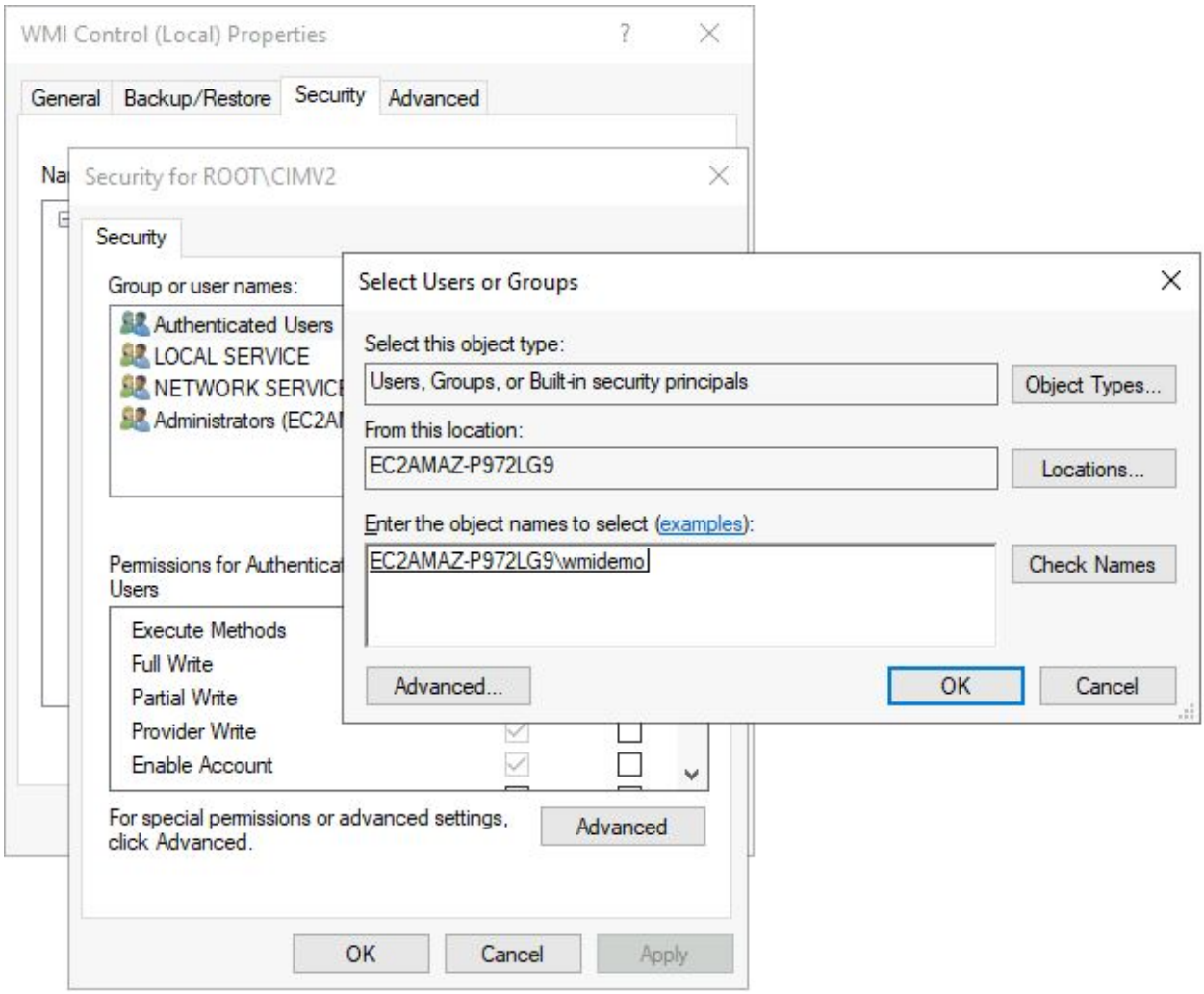




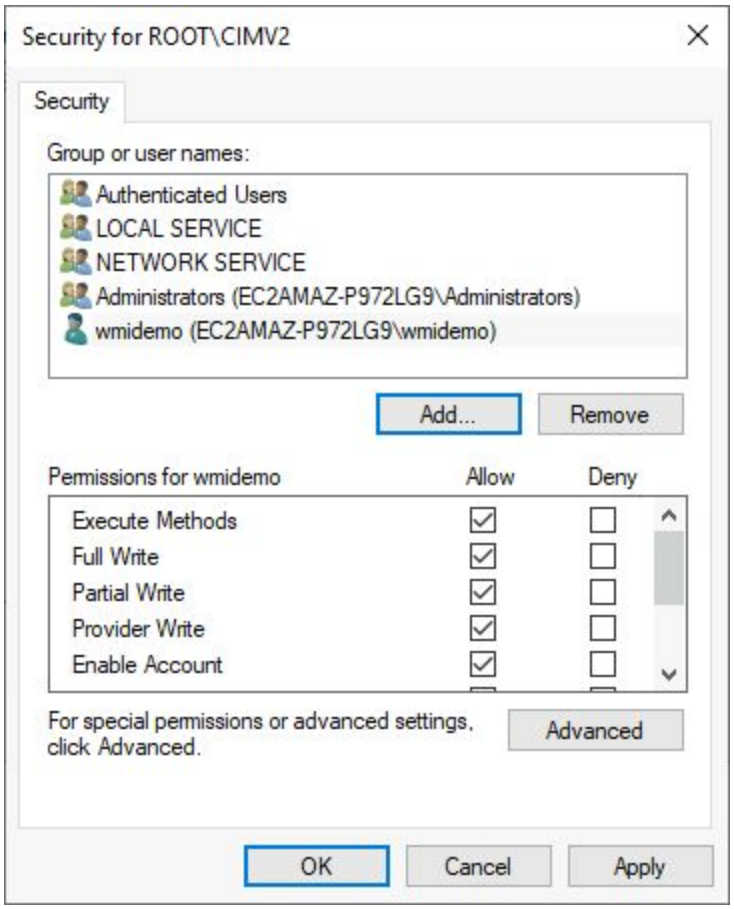
Click on "**Security**" and expand the "**Root**" tree and navigate "**CIMV2**" namespace



Click on below "**Security**" button and add the wmidemo user.



Give all the permissions to the user.



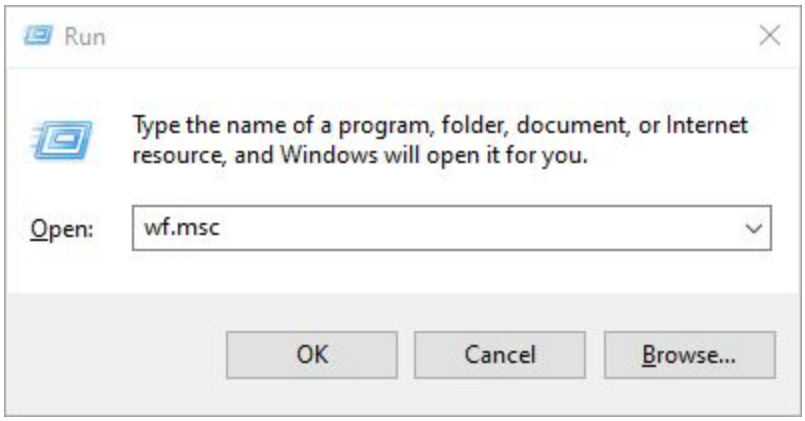
Click "**Apply**" and "**Ok**"

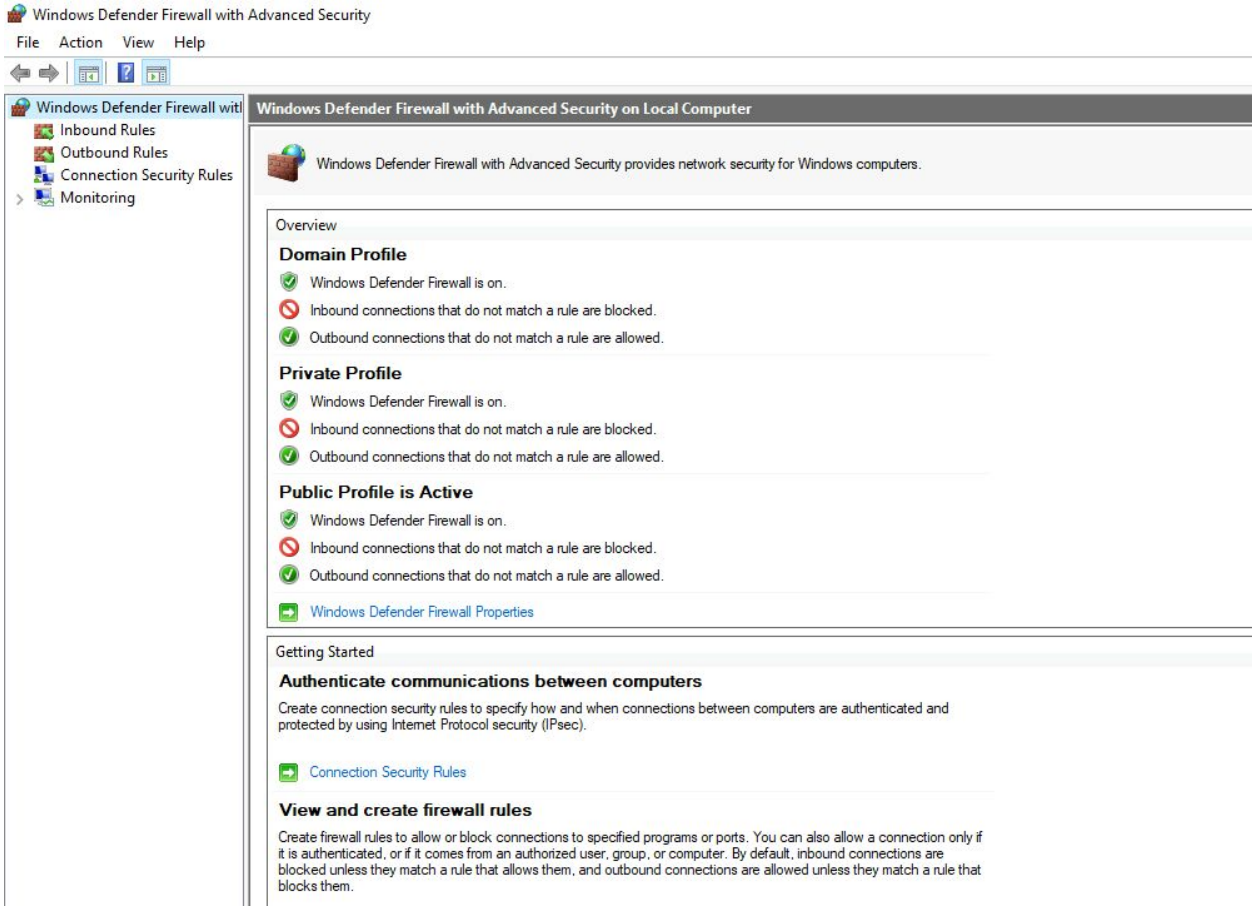
We have successfully configured the **wmidemo** user for WMI remote access.

Configure Firewall Rules

Note: Before we apply the WMI firewall rule we should keep in mind that WMI uses TCP port 135 and a range of dynamic ports TCP 49152-65535. So, we need to make sure that higher range ports are allowed too.

Step 1: Configure the Firewall rules to allow incoming connections for WMI service. Press Windows + R, type **wf.msc** in the Run dialog, and hit the Enter key to open it.





Step 2: Expand the Menu tree as follows: I**nbound Rules** and enable the following three rules:

- **Windows Management Instrumentation (ASync-In)**
- **Windows Management Instrumentation (DCOM-In)**
- **Windows Management Instrumentation (WMI-In)**

Right-click and enable the rules.

Windows Management Instrumentation (ASync-In)	Windows Management Instrumentation (...	All	Yes	Allow
Windows Management Instrumentation (DCOM-In)	Windows Management Instrumentation (...	All	Yes	Allow
Windows Management Instrumentation (WMI-In)	Windows Management Instrumentation (...	All	Yes	Allow
Windows Media Player (UDP-In)	Windows Media Player	All	No	Allow
Windows Media Player x86 (UDP-In)	Windows Media Player	All	No	Allow

We have enabled all necessary WMI firewall rules.

Step 3: Invoke the command cmdlet and check running processes.

Note: Please check the remote server i.e **Target Machine** IP address. By running the "**ipconfig**" command.

Administrator: Windows PowerShell

```
PS C:\> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : ap-southeast-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::a9ad:552c:7835:2507%6
   IPv4 Address. . . . . . . . . . . : 10.0.0.182
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1
PS C:\>
```

Execute Commands on a Remote Server

Switch to the "**Attacker Machine**" and execute the query using PowerShell **Get-WmiObject**.

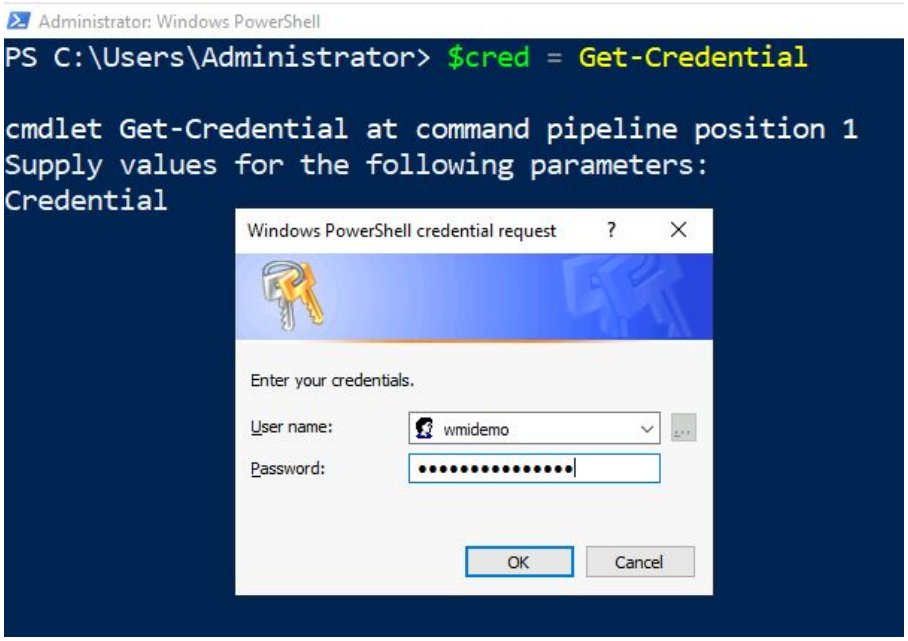
Step 1: We will use the Get-WMIObject cmdlet to execute commands on the remote server and we will fetch basic information about the target server.

Note: We have only given **root/cimv2** namespace permission to the user (wmidemo) so we won't be able to execute or get any information out of these namespace classes. The **root/cimv2** contains the 277 classes.

List of CIMV2 Classes: https://powershell.one/wmi/root/cimv2

Store the target machine credentials in the $cred variable:

Command: $cred = Get-Credential



Step 2: Get the processor information of the target machine.

Command: Get-WmiObject Win32_Processor -ComputerName 10.0.0.182 -Credential $cred

Attacker Machine Target Machine

Administrator: Windows PowerShell

```
PS C:\Users\Administrator> Get-WmiObject Win32_Processor -ComputerName 10.0.0.182 -Credential $cred


Caption           : Intel64 Family 6 Model 79 Stepping 1
DeviceID          : CPU0
Manufacturer      : GenuineIntel
MaxClockSpeed     : 2300
Name              : Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
SocketDesignation : CPU 1



PS C:\Users\Administrator>
```

We have received processor information. The WMI configuration is good to go.

References:

- https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/service-overview-and-network-port-requirements
- https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-wmiobject?view=powershell-5.1#:~:text=The%20Get%2DWmiObject%20cmdlet%20gets,available%20in%20a%20specified%20namespace.
- https://docs.microsoft.com/en-us/windows/win32/wmisdk/connecting-to-wmi-on-a-remote-computer-by-using-powershell