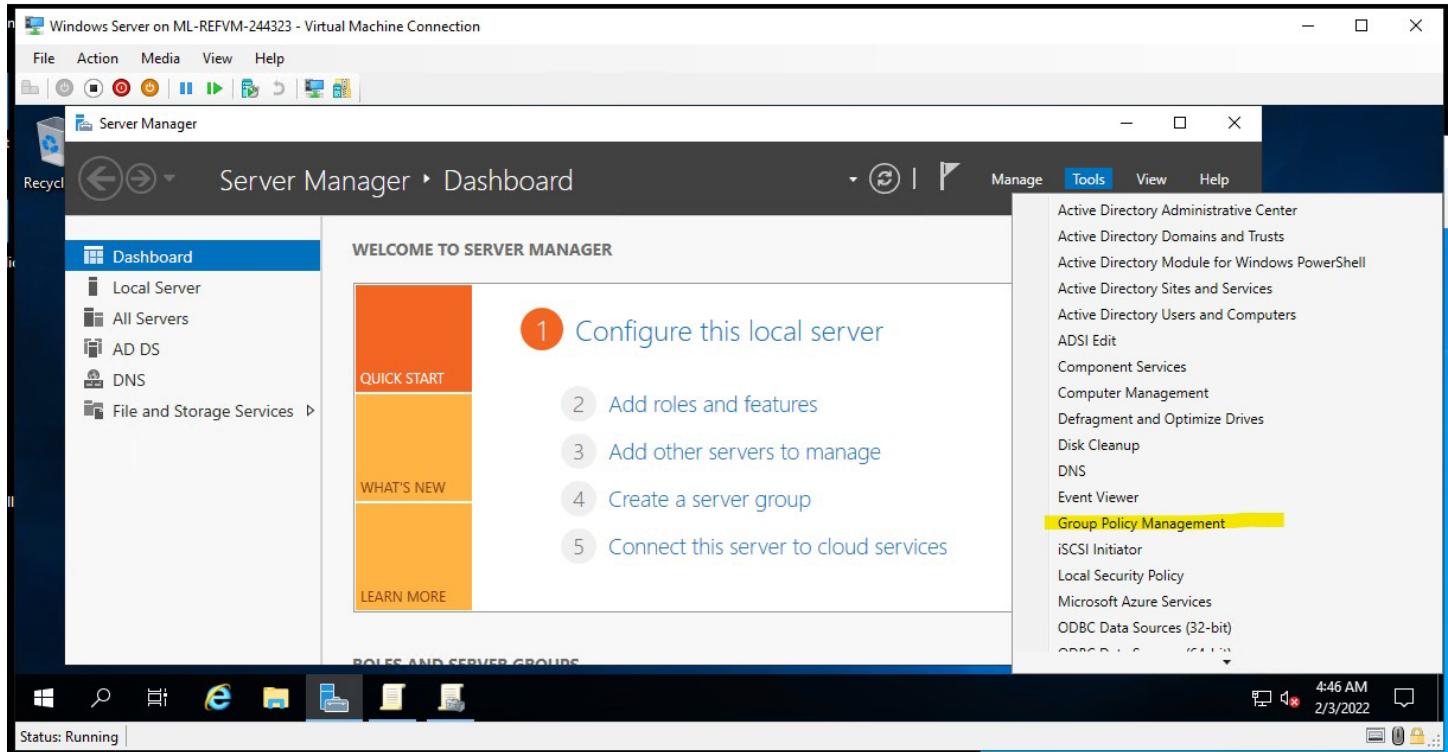
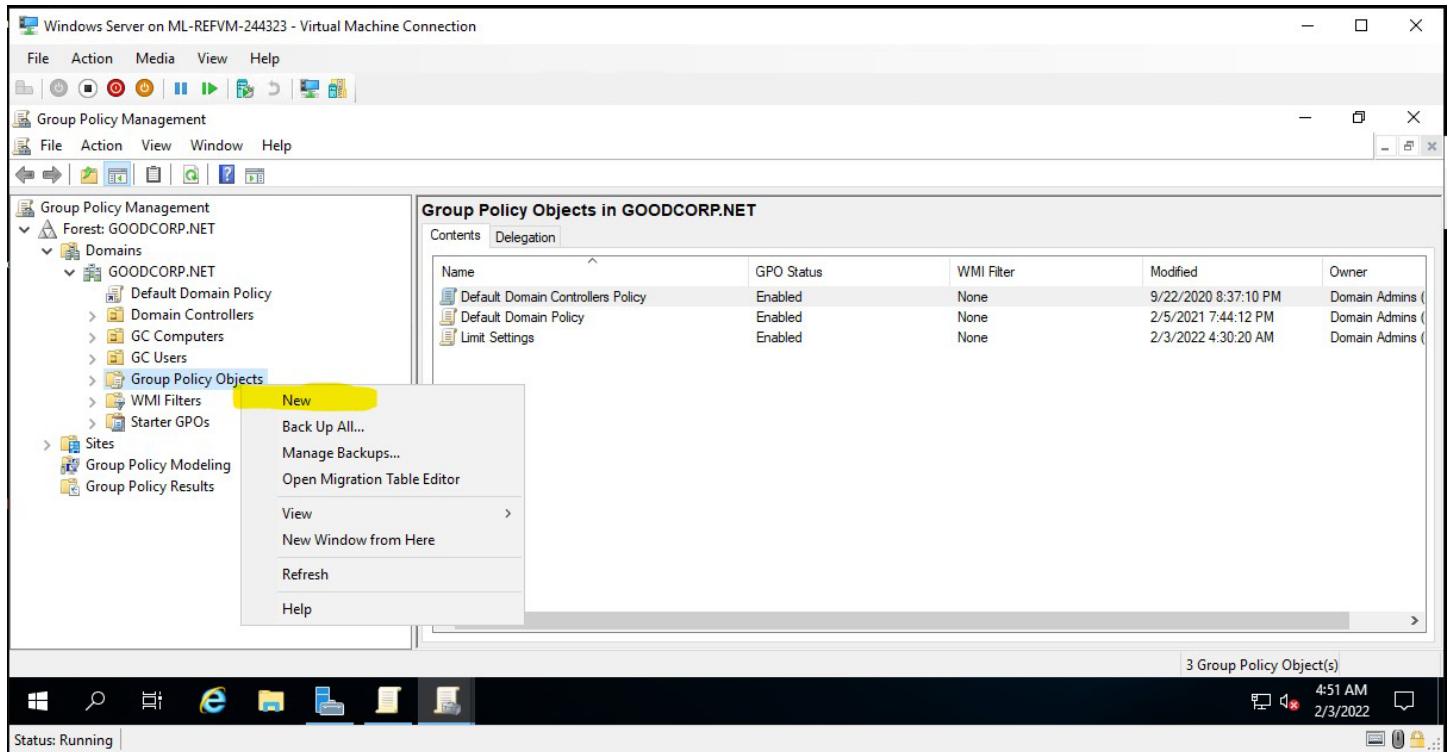


Task 1: Create a GPO: Disable Local Link Multicast Name Resolution

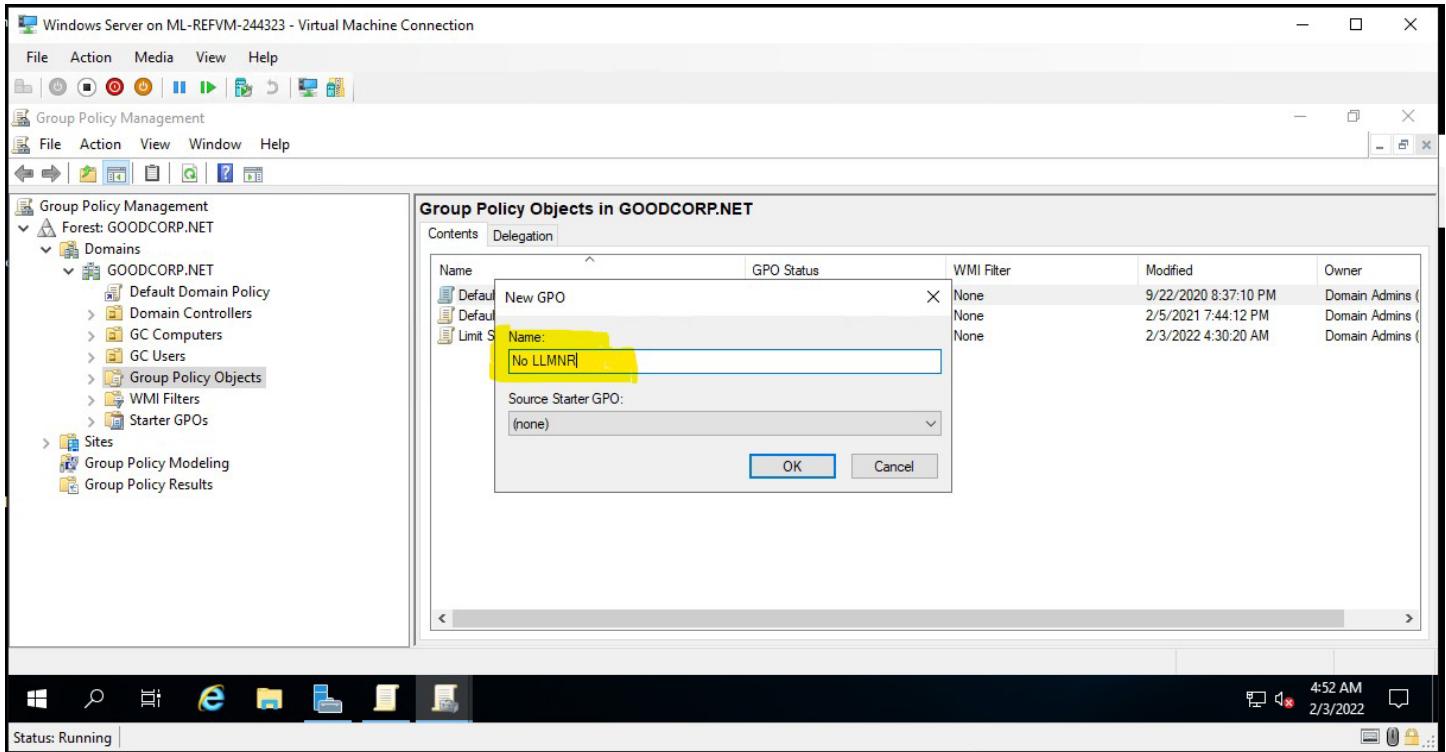
- On the top-right of the Server Manager screen, open the **Group Policy Management** tool to create a new **GPO**.



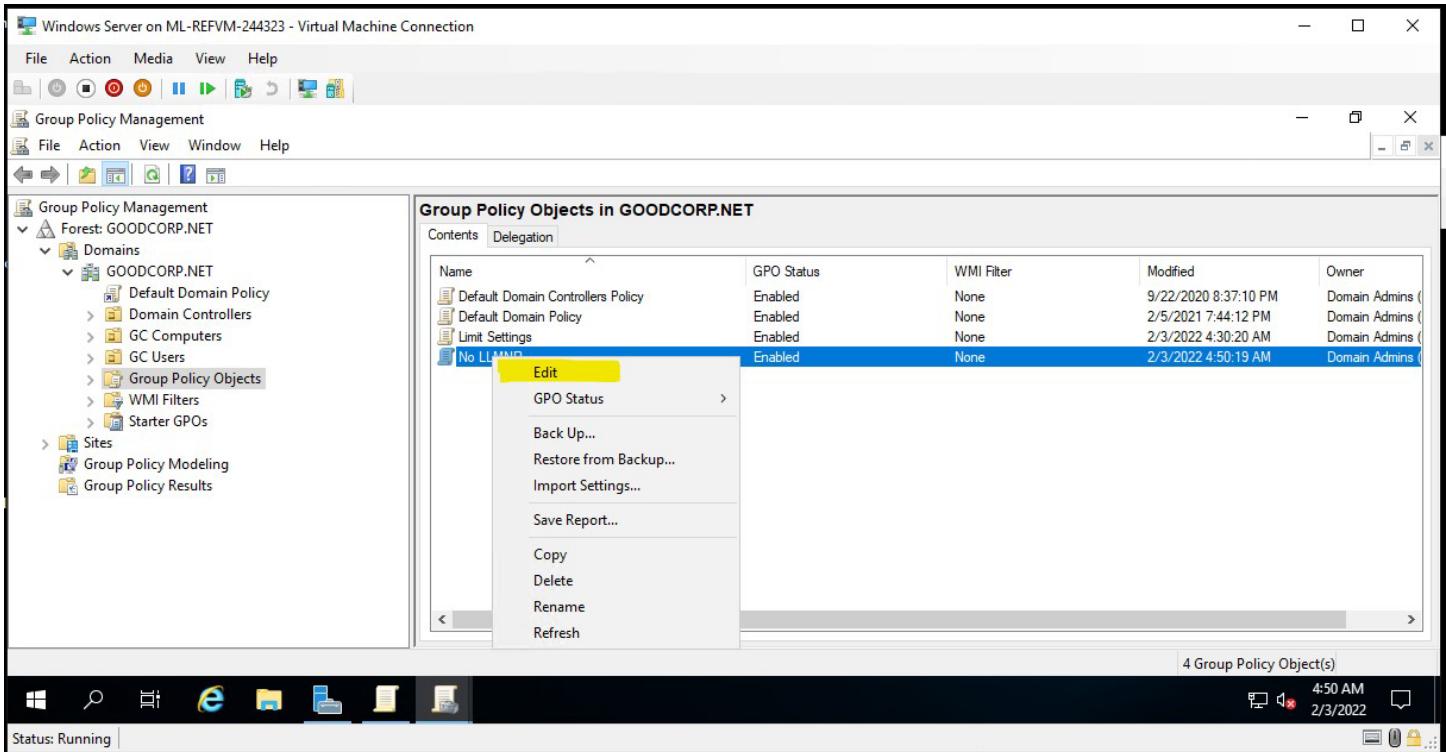
- Right-click **Group Policy Objects** and select **New**.



3. Name the Group Policy Object **No LLMNR**.



4. Right-click the new **No LLMNR** GPO listing and select **Edit** to open the **Group Policy Management Editor** and find policies.

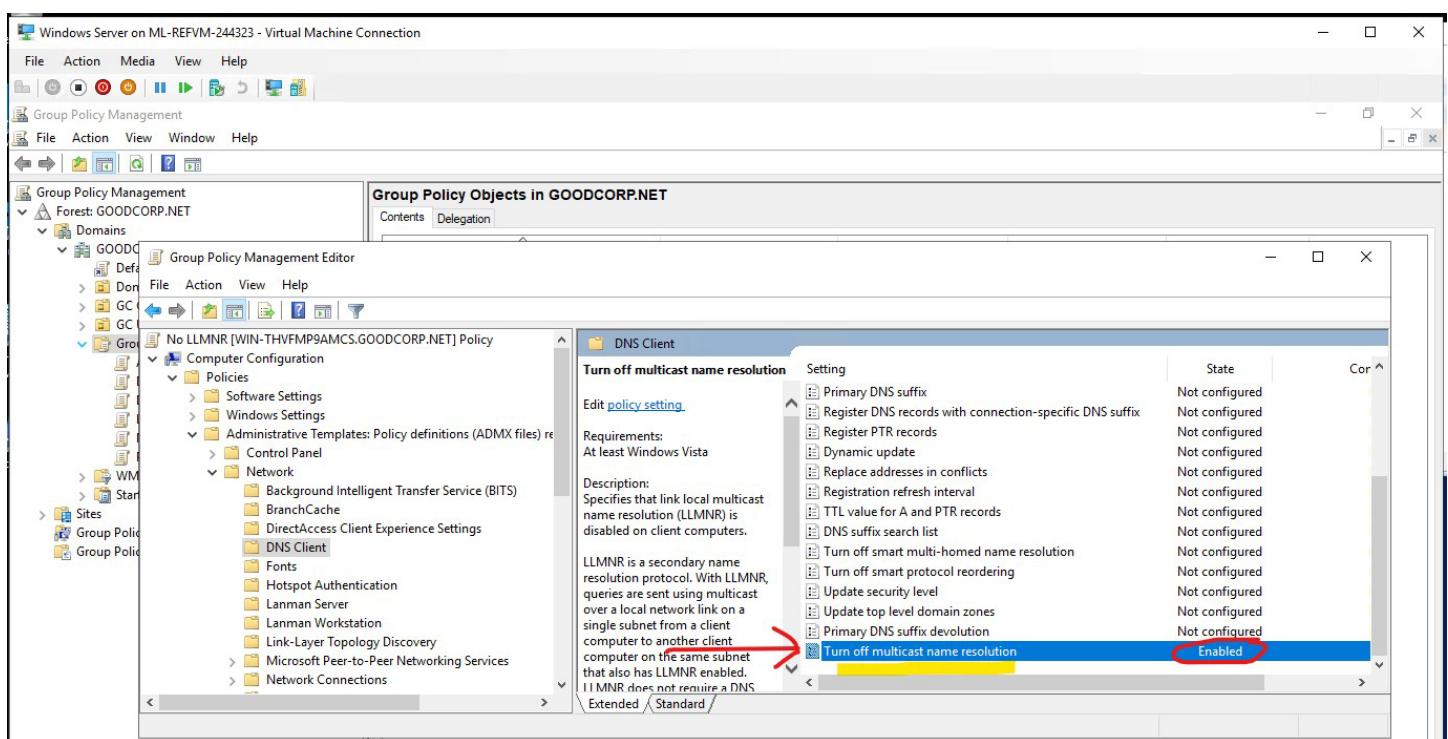
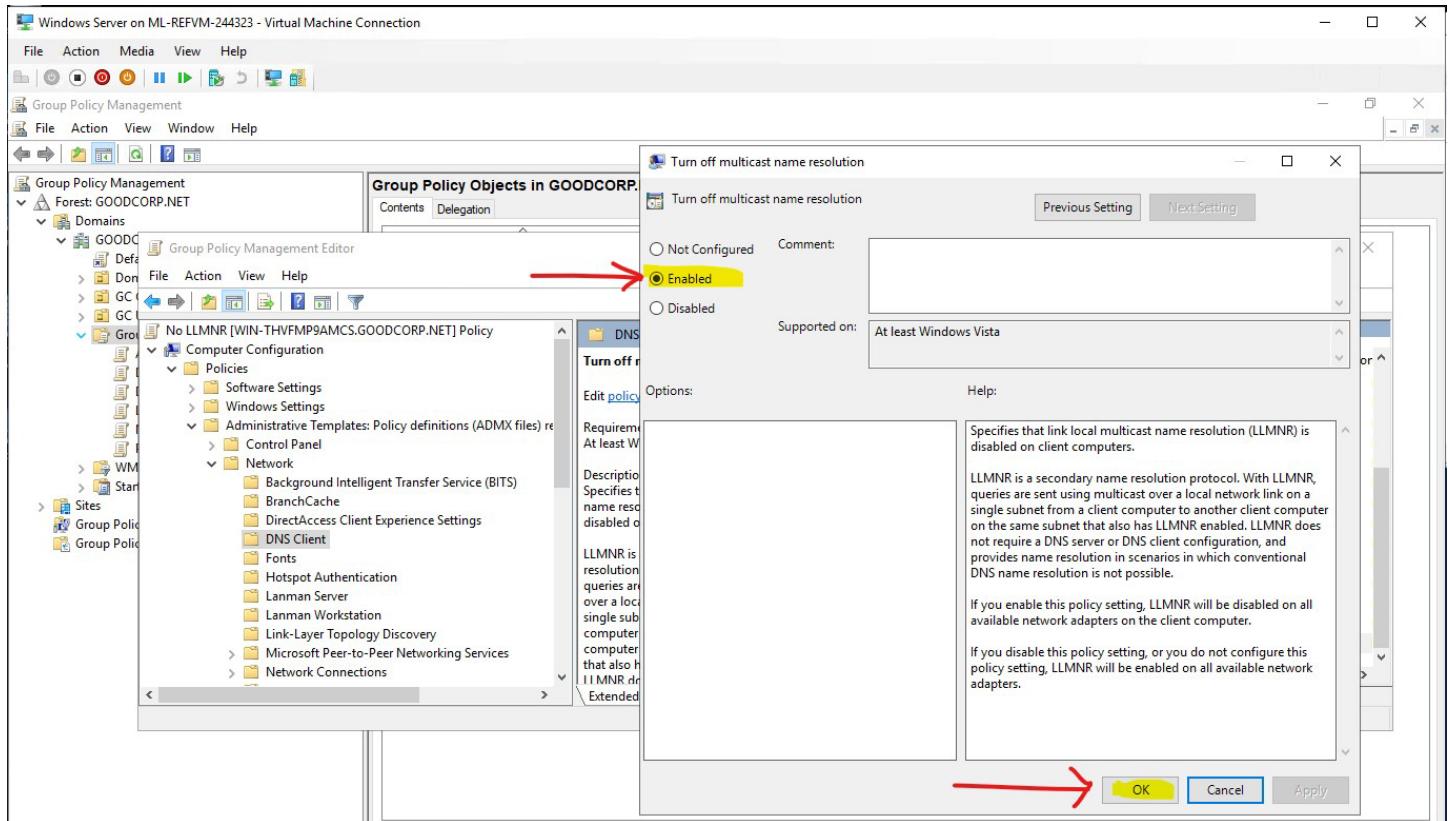


5. In the Group Policy Management Editor, the policy you are looking for is at the following path:

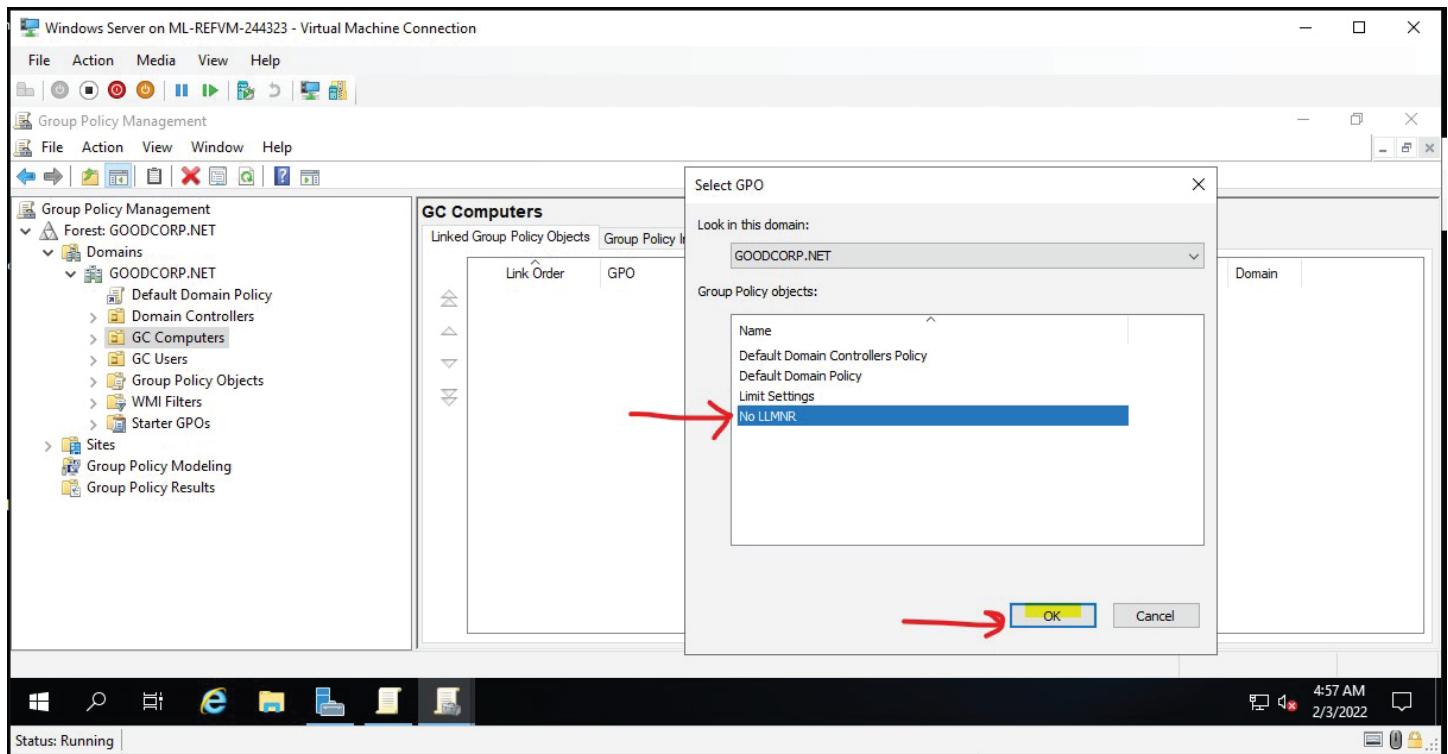
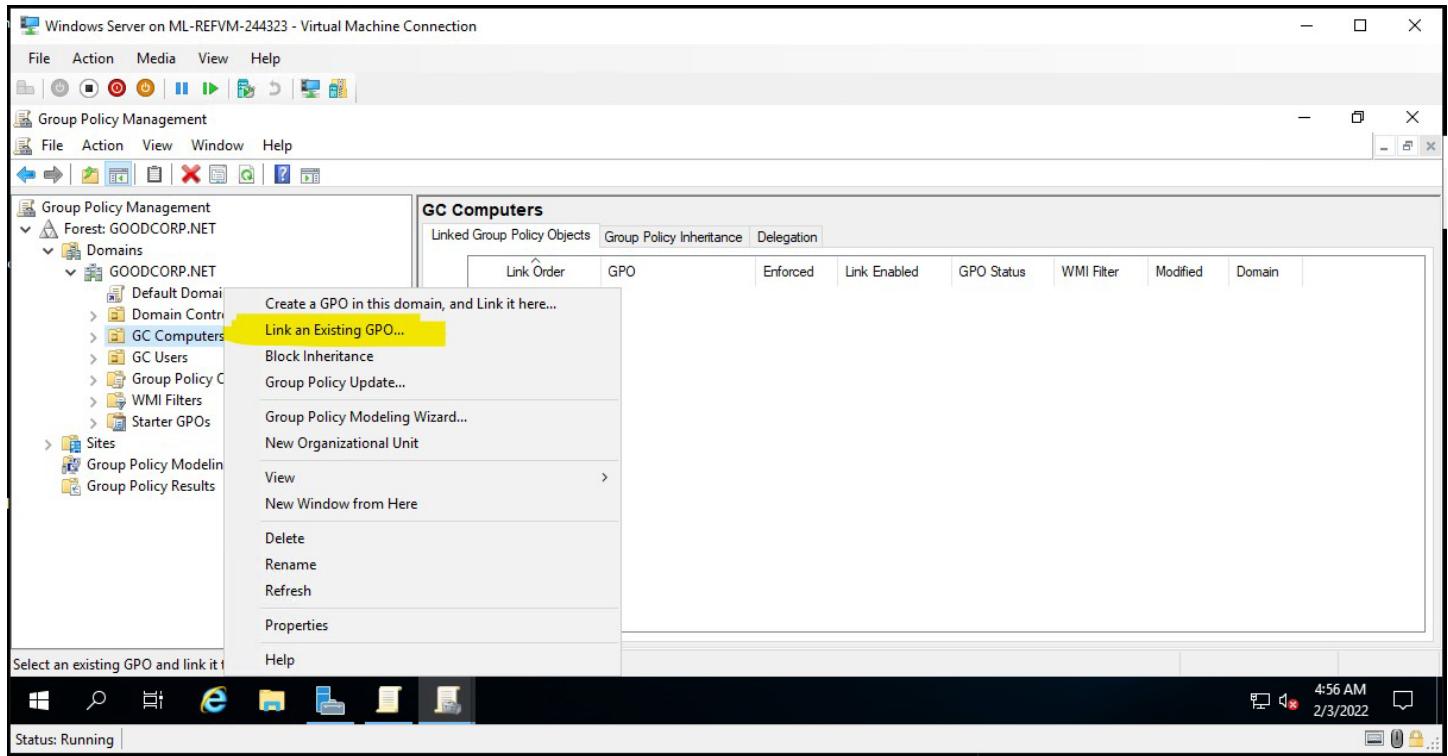
Computer Configuration\Policies\Administrative Templates\Network\DNS Client .

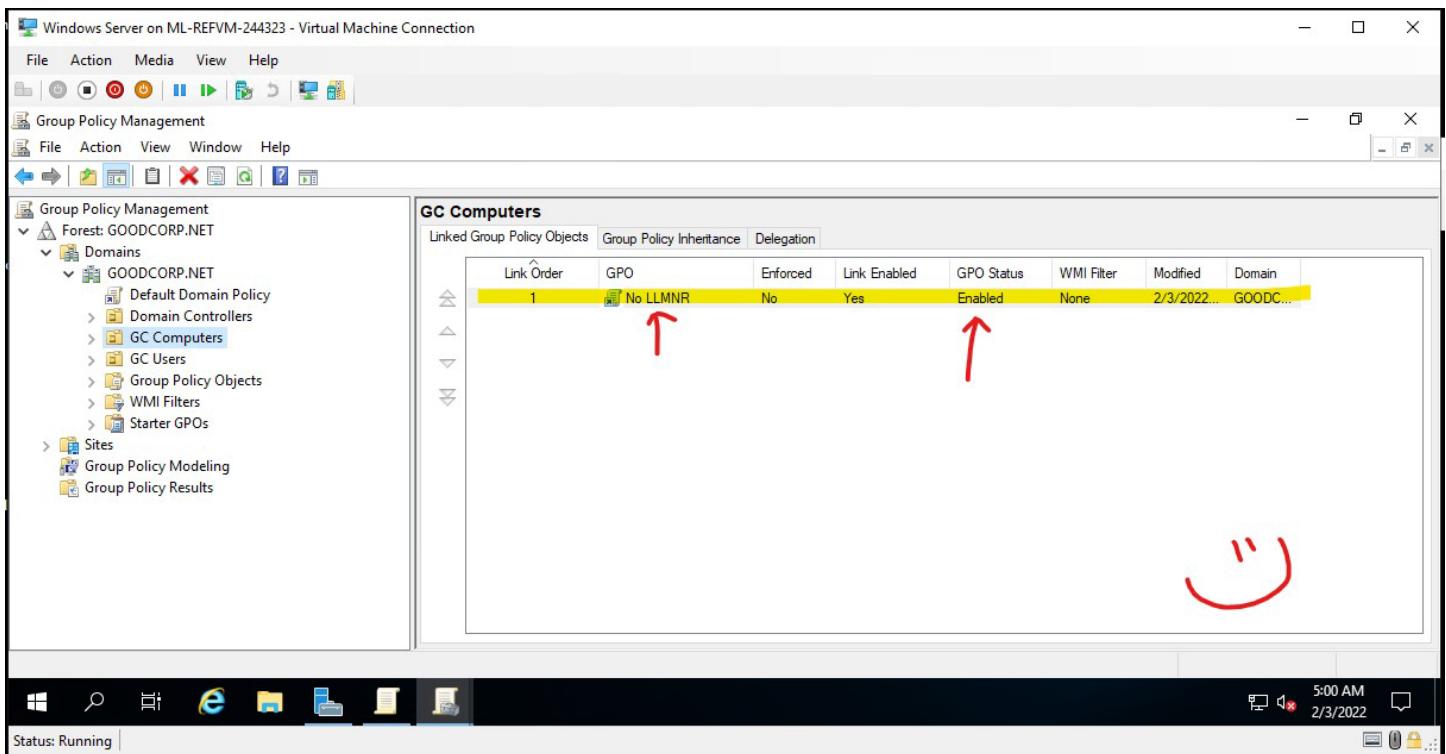
-Find the policy called **Turn Off Multicast Name Resolution** .

-Enable this policy.



6. Exit the **Group Policy Management Editor** and link the GPO to the **GC Computers** organizational unit you previously created.



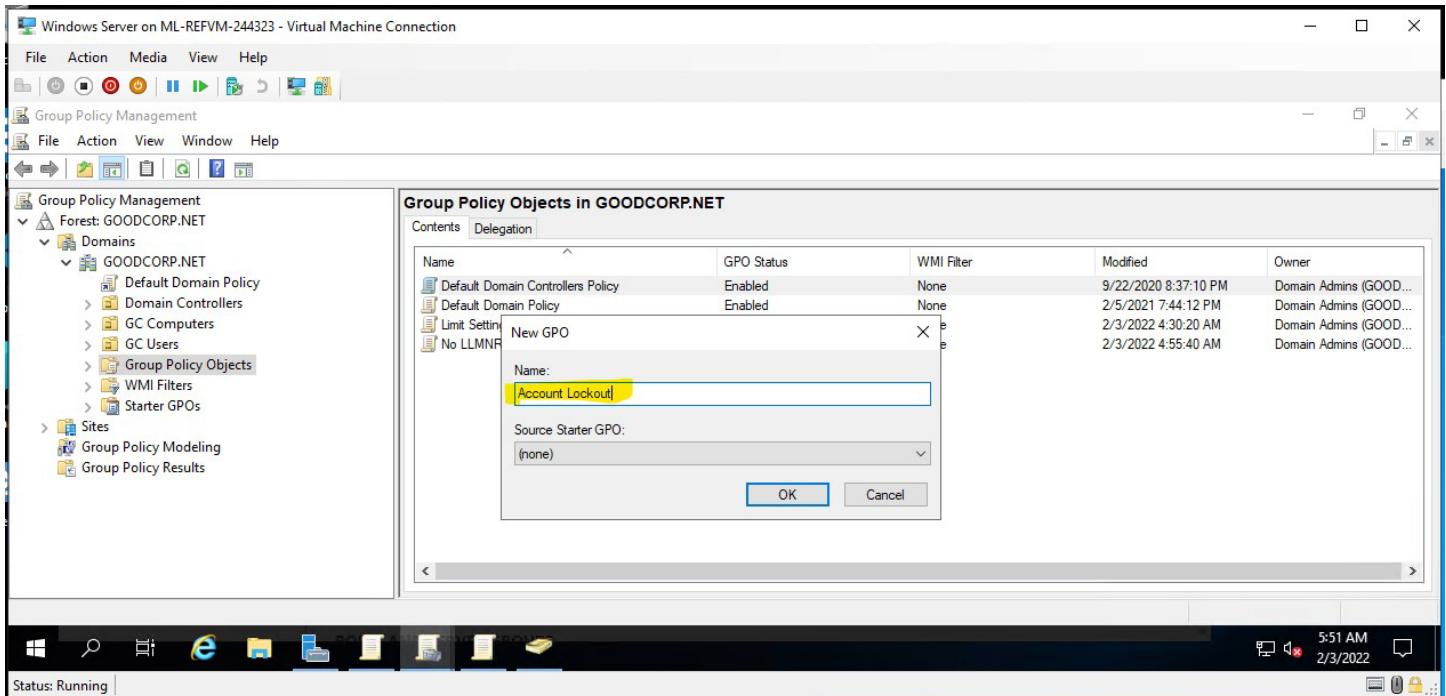


Task 2: Create a GPO: Account Lockout

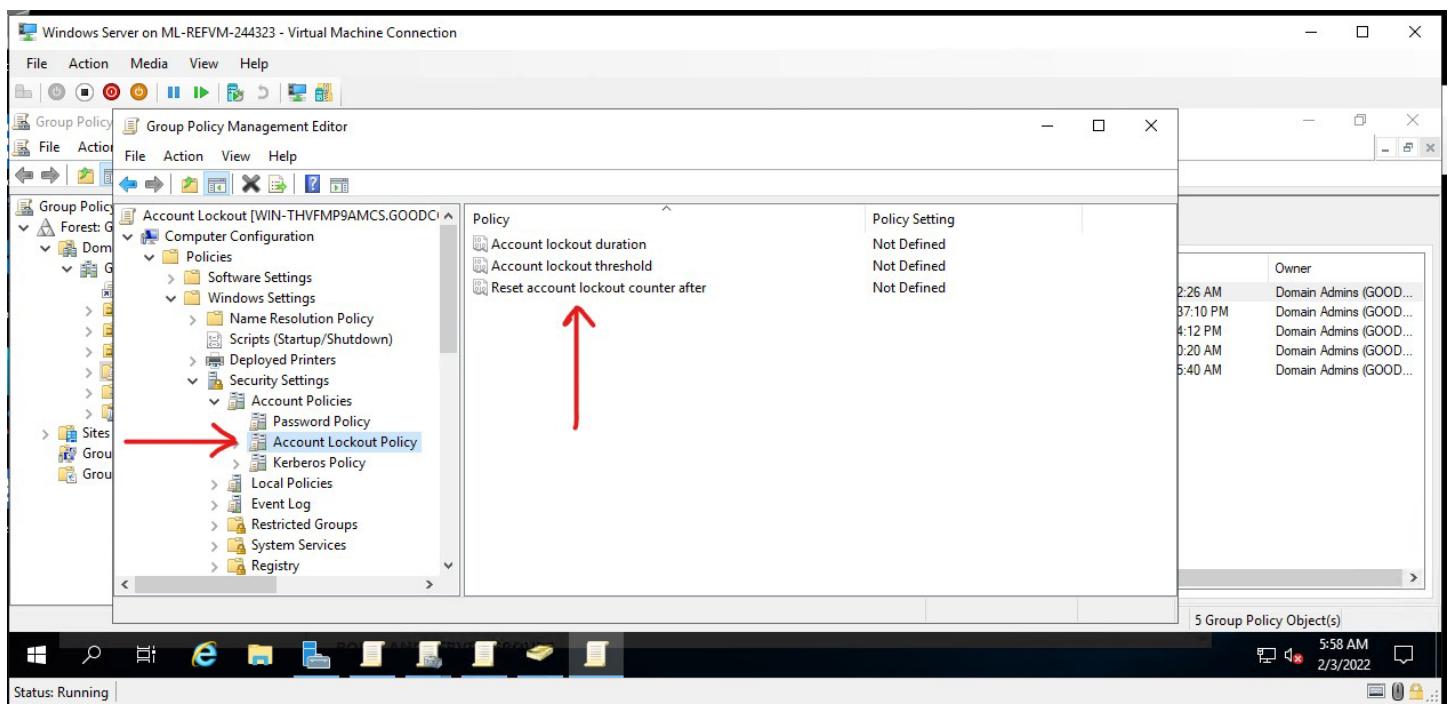
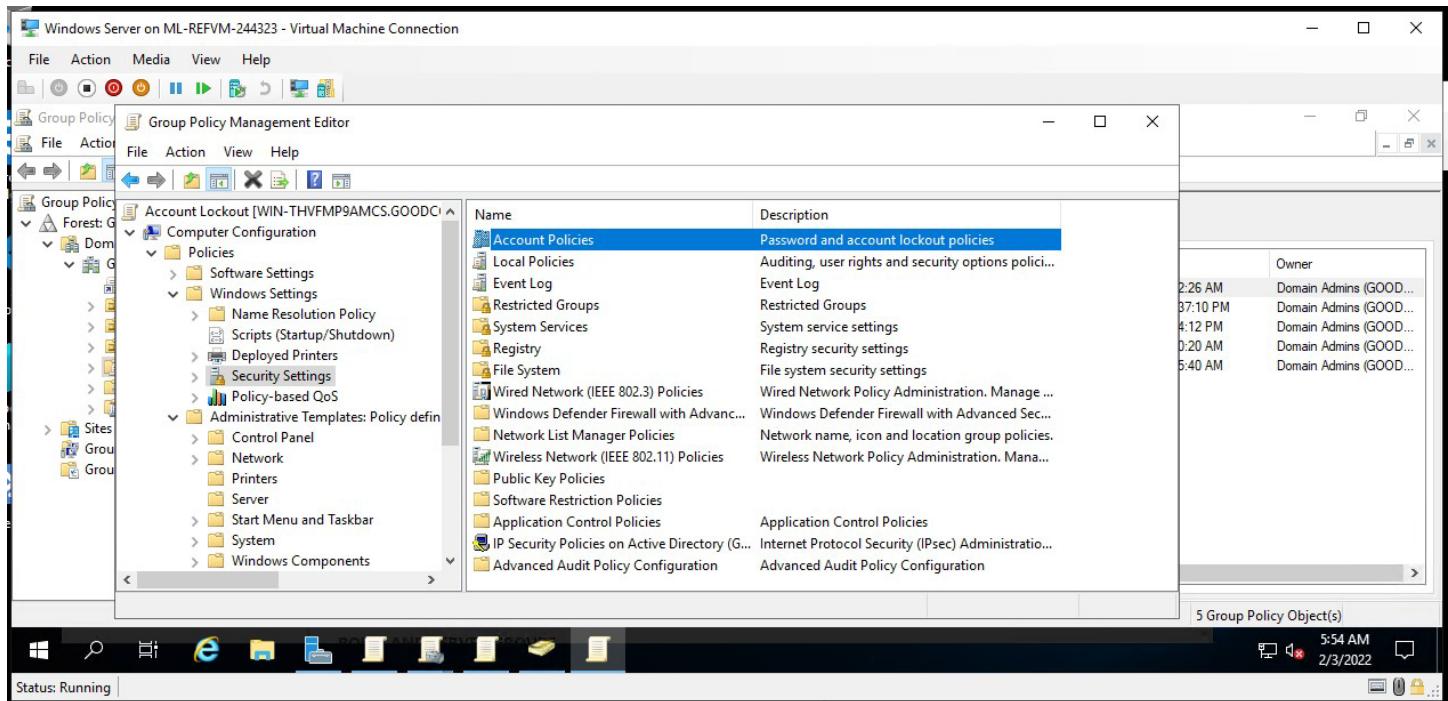
You'll be working within in your nested Windows Server machine again to create another **Group Policy Object**.

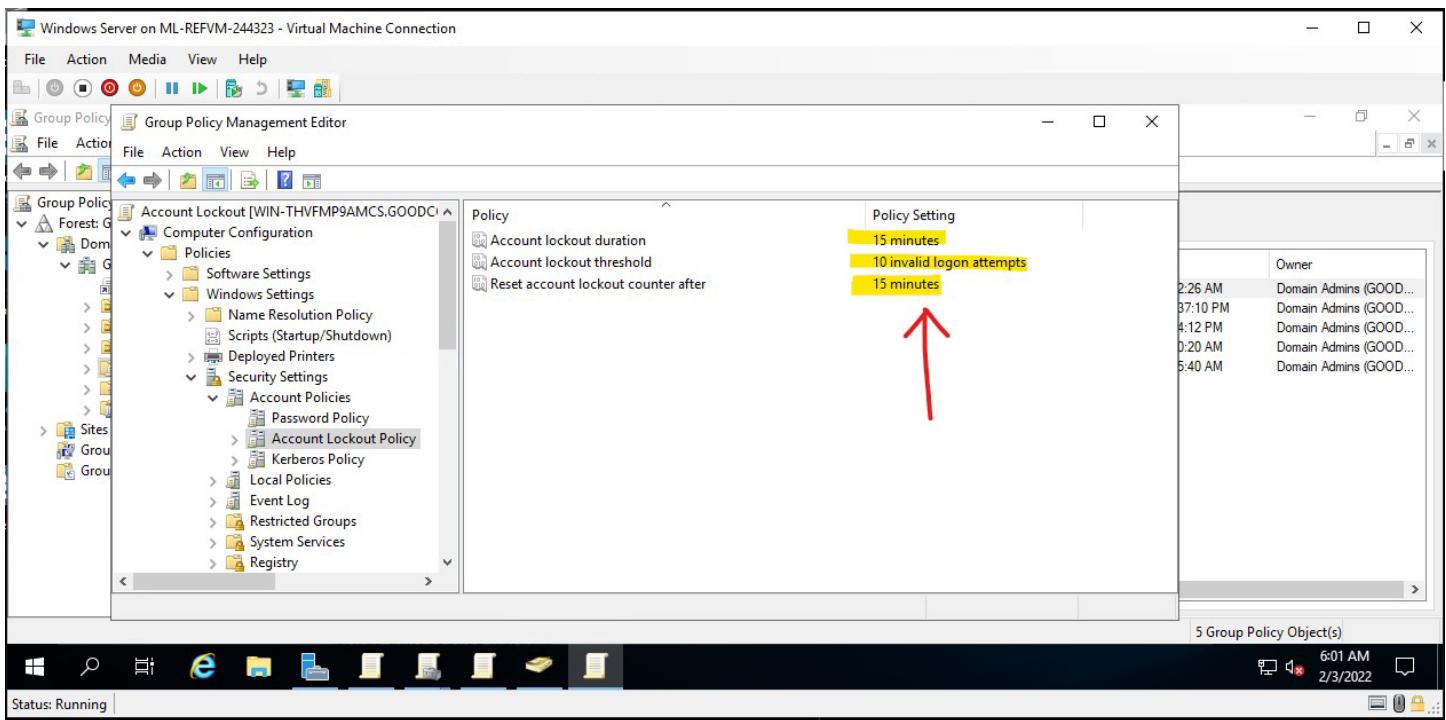
Create what you believe to be a reasonable account lockout Group Policy for the Windows 10

1. Name the Group Policy Object **Account Lockout**.



2. You can use Microsoft's 10/15/15 recommendation if you'd like.

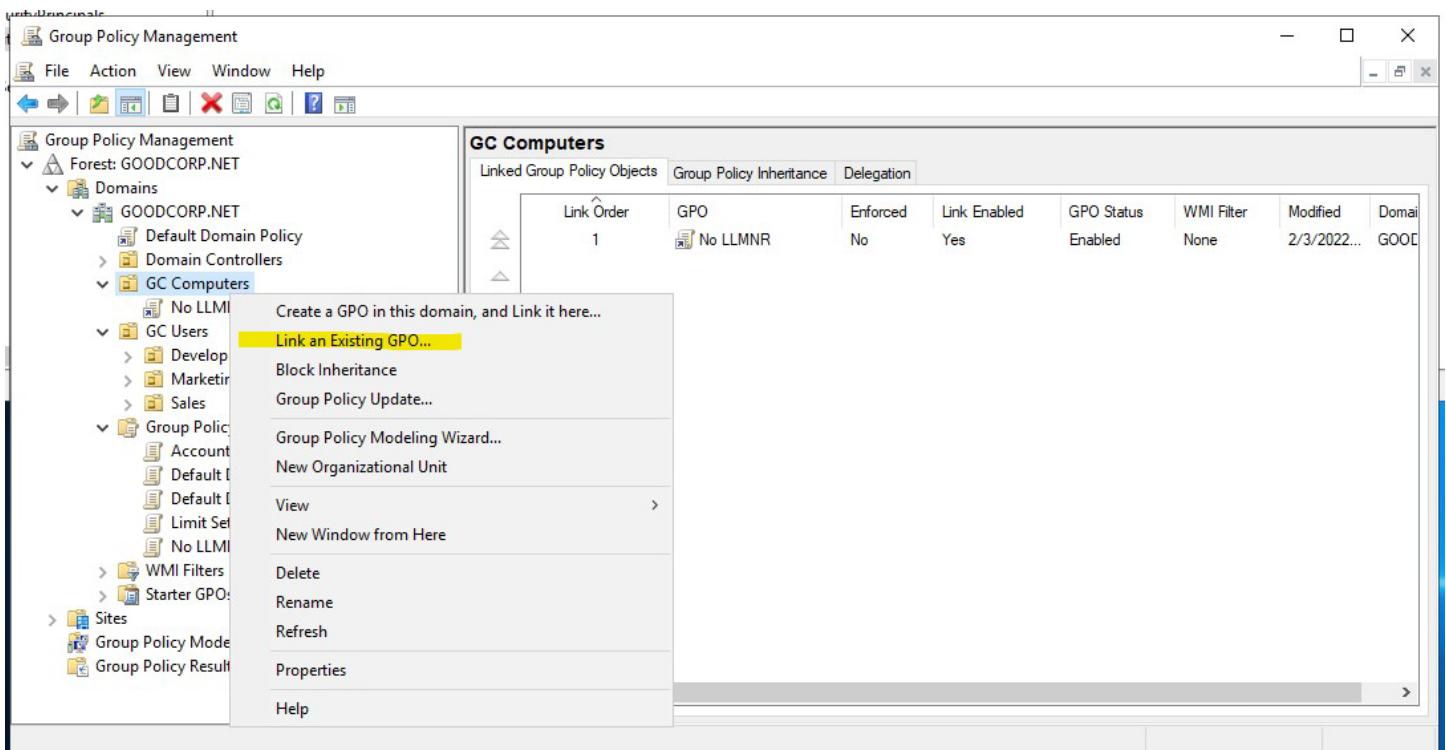


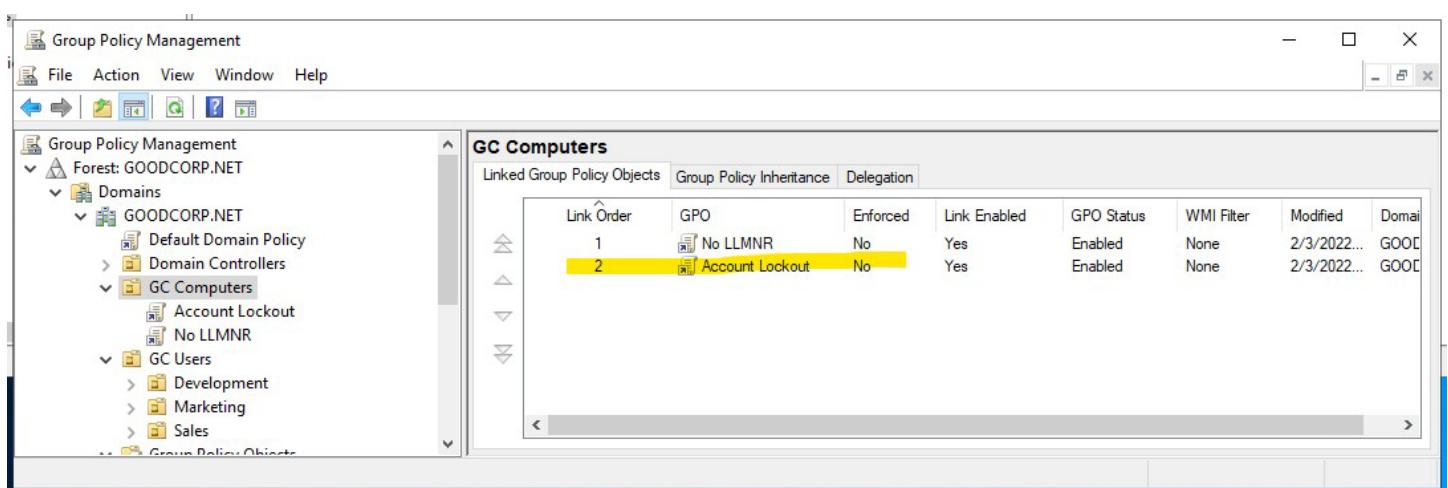
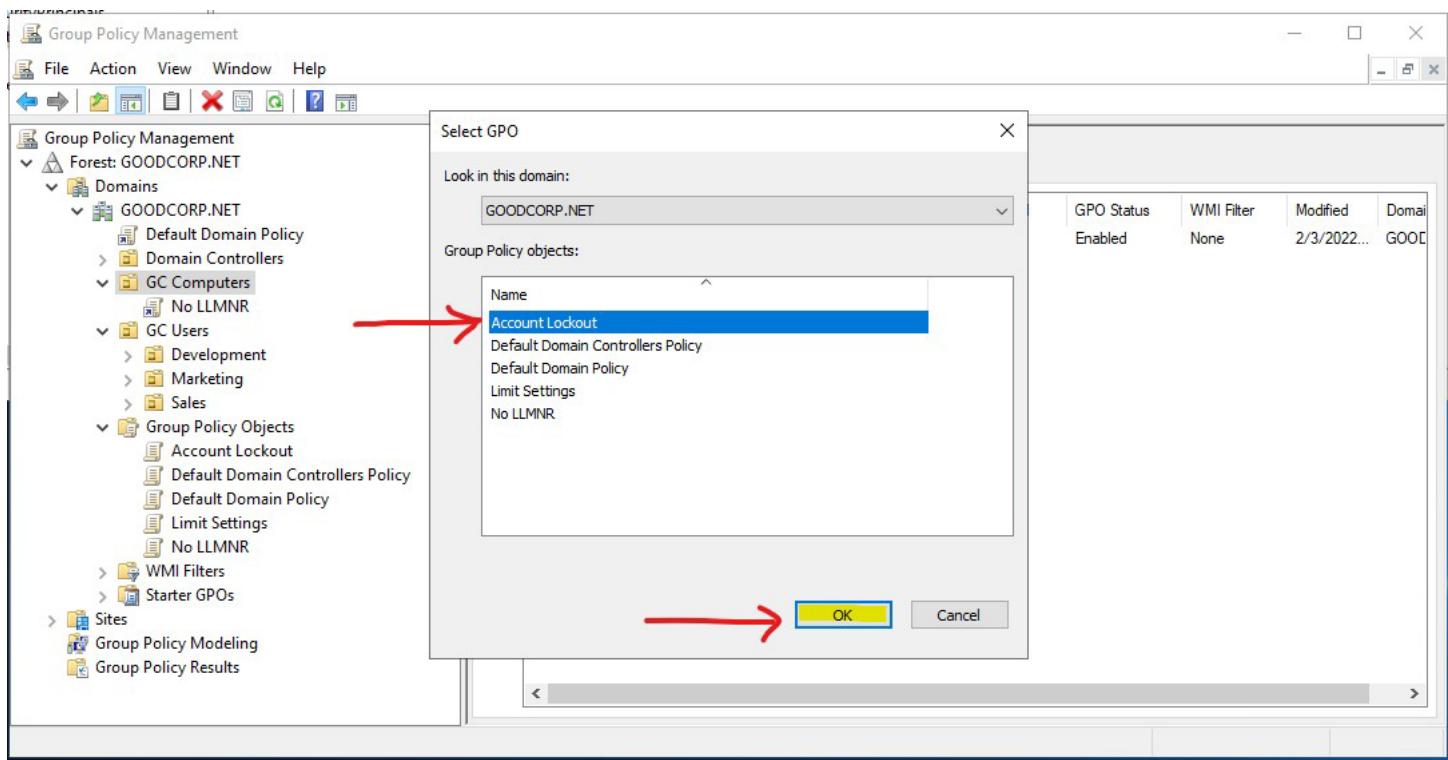


3. When editing policies for this new GPO, keep in mind that you're looking for **computer configuration** policies to apply to your **GC Computers** OU. Also, these policies involve **Windows security settings** and **accounts**.

Yep, should have told me in #2 but I found it. :)

4. Don't forget to link the GPO to your **GC Computers** organizational unit.

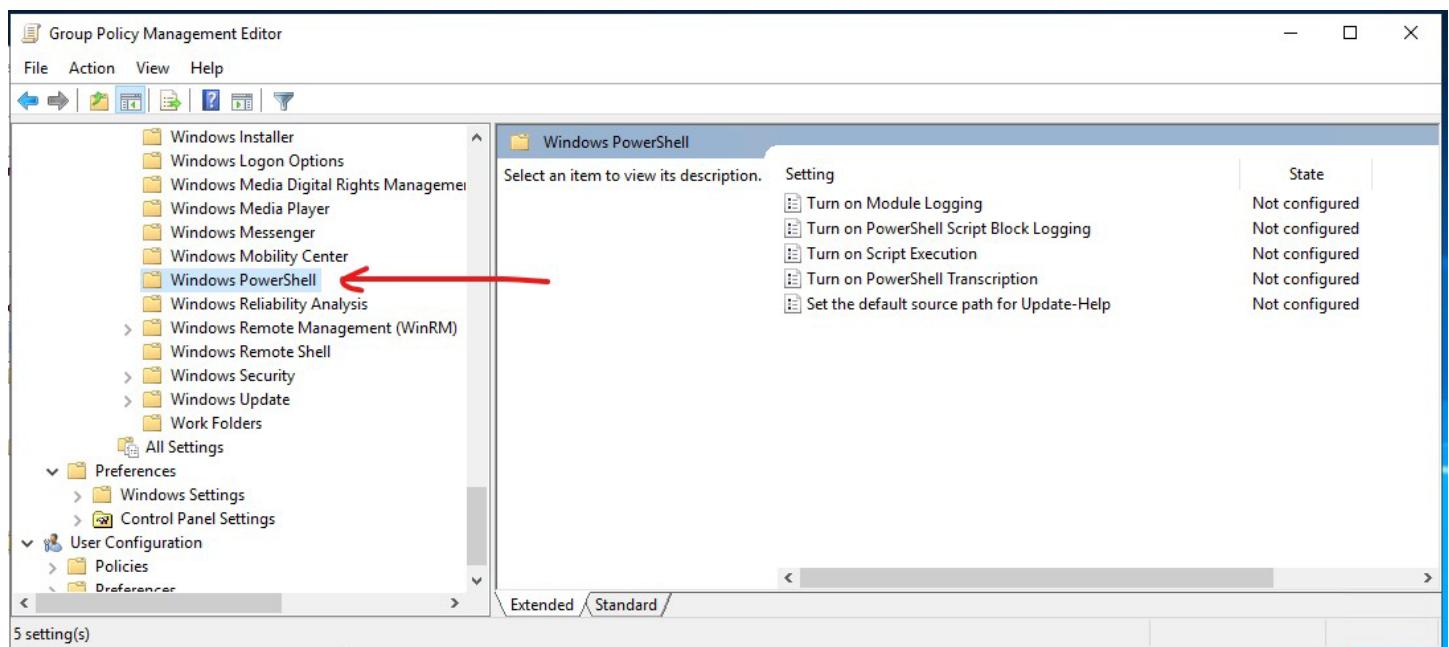
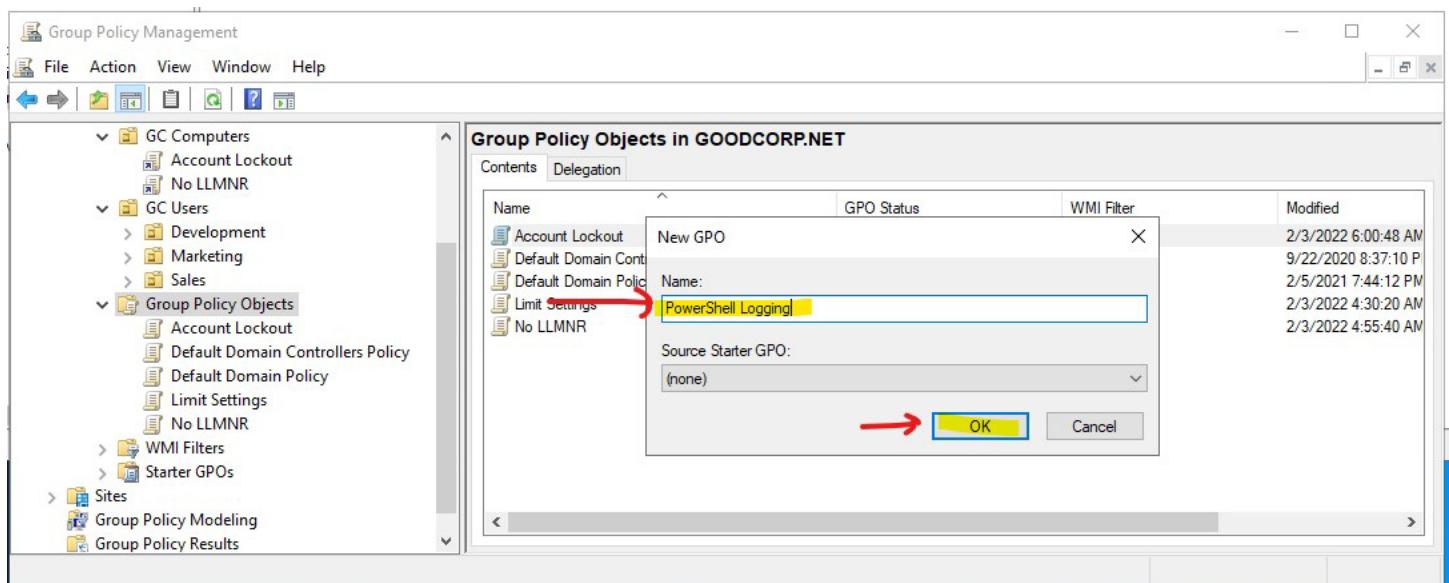




Task 3: Create a GPO: Enabling Verbose PowerShell Logging and Transcription

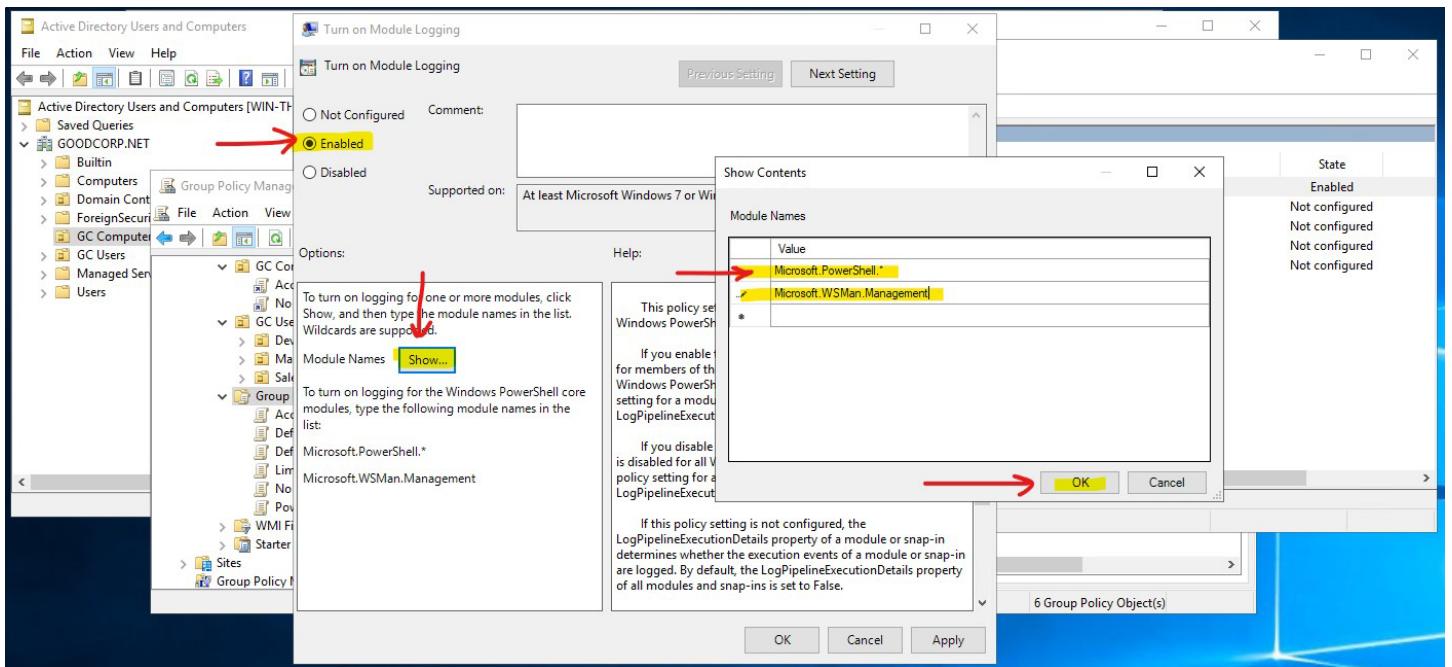
1. Name the Group Policy Object **PowerShell Logging**.

- Find the proper Windows Powershell policy in **Group Policy Management Editor**.
- Hint: Check out the **computer configuration, administrative templates, and Windows component directories**.



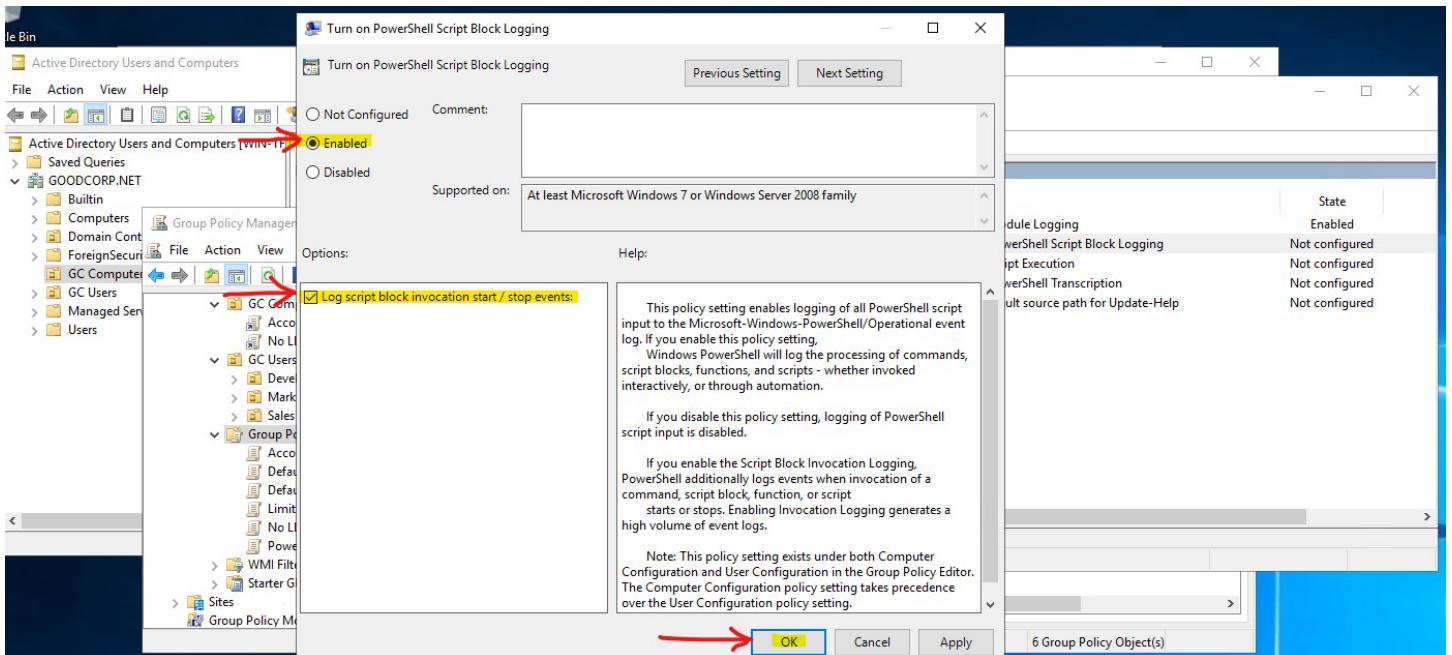
2. Enable the ***Turn on Module Logging*** and do the following:

- Click **Show** next to **Module Names**.
- Since we want to log **all** PowerShell modules, enter an asterisk * (wildcard) for the **Module Name**, then click **OK**.



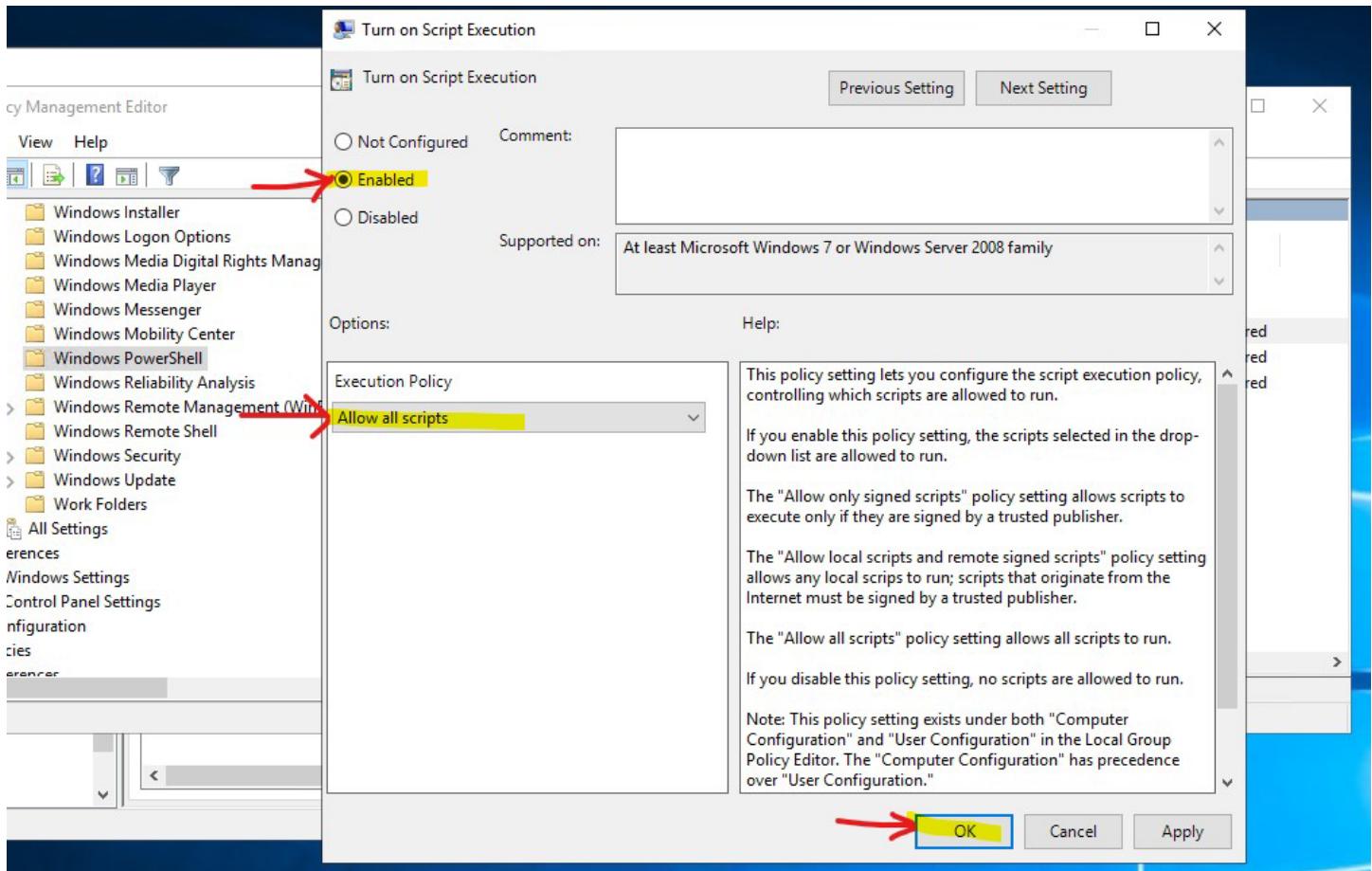
3. Enable the ***Turn on PowerShell Script Block Logging*** policy.

- Make sure to check the **Log script block invocation start/stop events:** setting.



4. Enable the ***Turn on Script Execution*** policy and do the following:

Set Execution Policy to Allow all scripts.



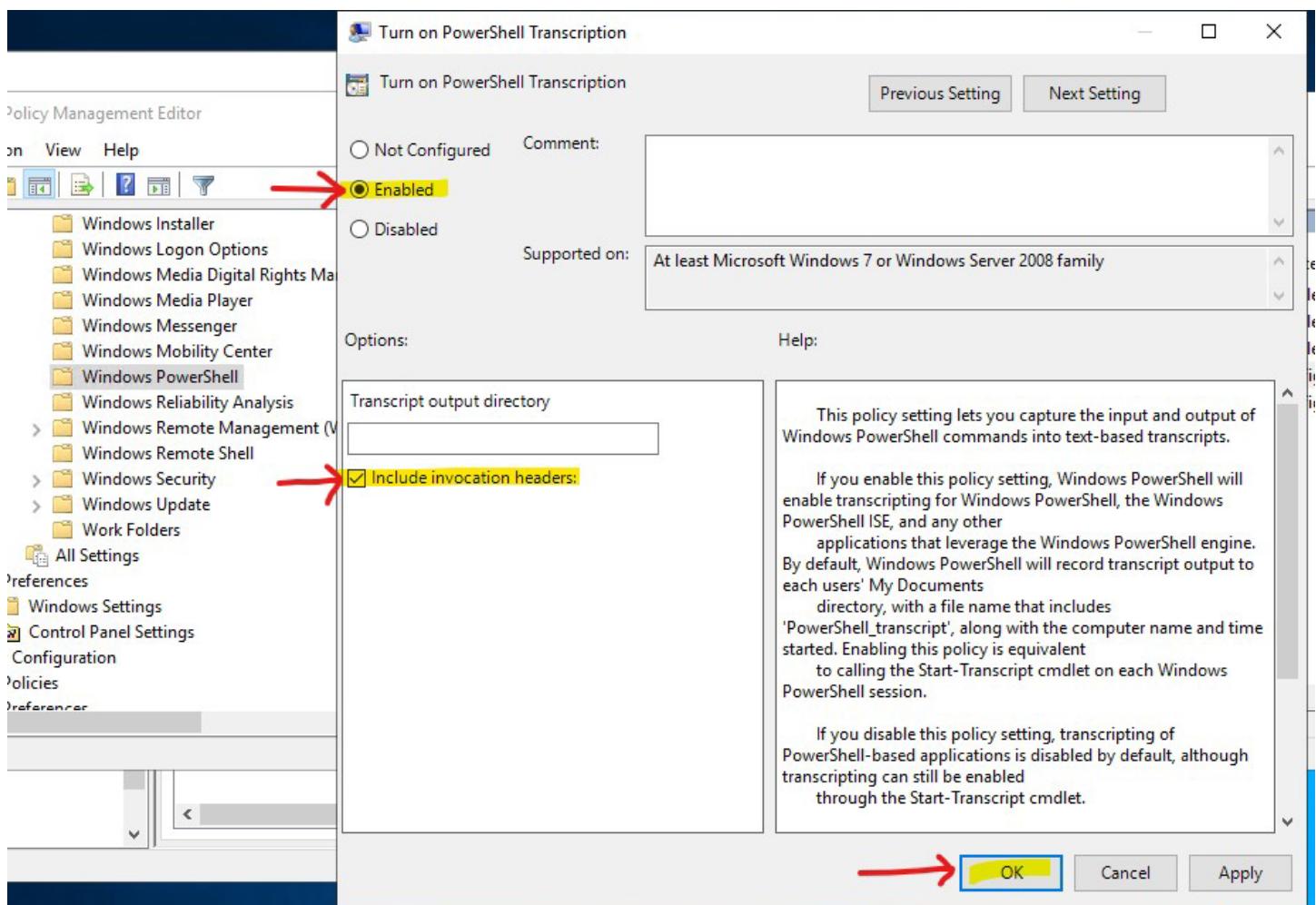
Note: Do you remember the `Set-ExecutionPolicy` cmdlet we ran during the PowerShell exercises?
This policy can enforce those settings as part of a GPO.

5. Enable the ***Turn on PowerShell Transcription*** policy and do the following:

-Leave the **Transcript output directory** blank (this defaults to the user's ~\Documents directory).

Note: "Transcription" means that an exact copy of the commands are created in an output directory.

-Check the **Include invocation headers** option. This will add timestamps to the command transcriptions.



6. Leave the **Set the default source path for Update-Help** policy as **Not configured**.

Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Enabled	No
Turn on Script Execution	Enabled	No
Turn on PowerShell Transcription	Enabled	No
Set the default source path for Update-Help	Not configured	No

7. Link this new **PowerShell Logging** GPO to the **GC Computers** OU.

Link Order	GPO	Enforced	Link Enabled	GPO Status	WMI Filter	Modified	Domain
1	No LLMNR	No	Yes	Enabled	None	2/3/2022...	GOOD
2	Account Lockout	No	Yes	Enabled	None	2/3/2022...	GOOD
3	PowerShell Logging	No	Yes	Enabled	None	2/3/2022...	GOOD

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\sysadmin> gpupdate
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.

PS C:\Users\sysadmin>

```

Task 4: Create a Script: Enumerate Access Control Lists

Familiarize yourself with the basics of **Get-Acls**:

- Get-Acl** without any parameters or arguments will return the security descriptors of the directory you're currently in.
- Get-Acl <filename>** will return the specific file's ACL. We'll need to use this for our task.

Create a PowerShell script that will enumerate the Access Control List of each file or subdirectory within the current working directory.

1. Create a **foreach** loop. You can use the following template:

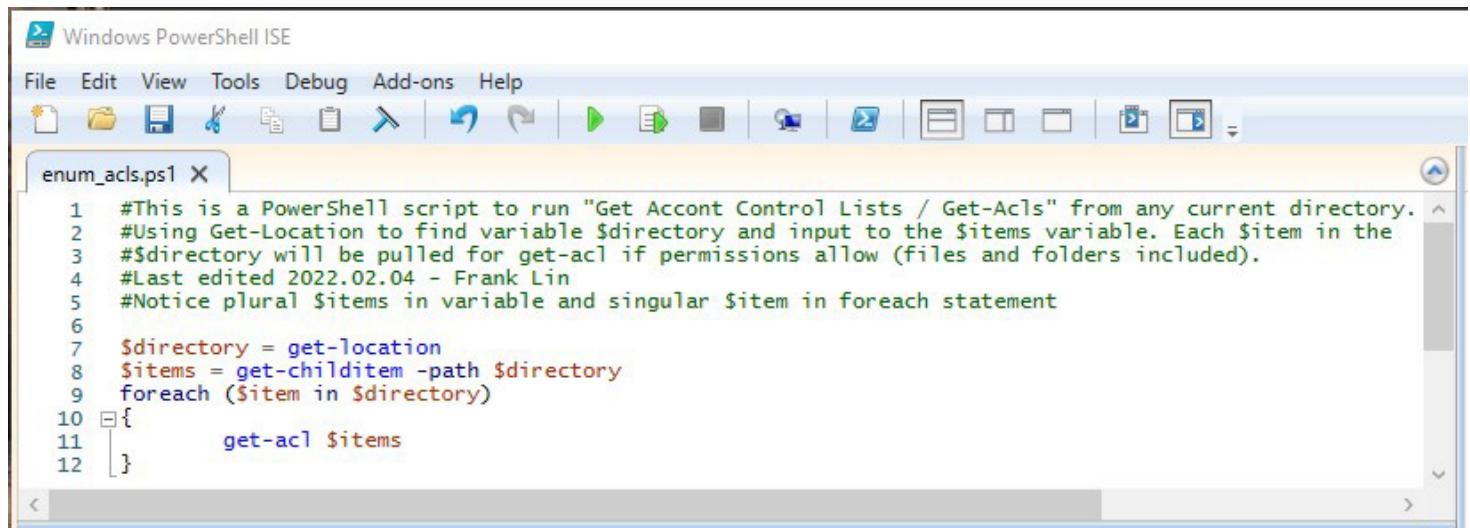
```
foreach ($item in $directory) {
    <Script block>
}
```

2. Above the **foreach** condition, set a variable, **\$directory**, to the contents of the current directory.

3. Replace the script block placeholder with the command to enumerate the ACL of a file, using the **\$item** variable in place of the file name.

You'll need to use the following cmdlets:

- Get-ChildItem** (or any alias of **Get-ChildItem**, such as **ls** or **dir**)
- Get-Acl**



The screenshot shows the Windows PowerShell ISE interface. The title bar says "Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, Help. The toolbar has various icons for file operations. The main code editor window contains the following PowerShell script:

```

enum_acls.ps1
1 #This is a PowerShell script to run "Get Accont Control Lists / Get-Acls" from any current directory.
2 #Using Get-Location to find variable $directory and input to the $items variable. Each $item in the
3 #$directory will be pulled for get-acl if permissions allow (files and folders included).
4 #Last edited 2022.02.04 - Frank Lin
5 #Notice plural $items in variable and singular $item in foreach statement
6
7 $directory = get-location
8 $items = get-childitem -path $directory
9 foreach ($item in $directory)
10 {
11     get-acl $items
12 }
```

4. Save this script in **C:\Users\sysadmin\Documents** as **enum_acls.ps1**.

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\sysadmin> cd .\Documents\
PS C:\Users\sysadmin\Documents> ls

    Directory: C:\Users\sysadmin\Documents

Mode                LastWriteTime         Length Name
----                -----          ----  -
d-----        2/5/2022   8:02 AM           20220205
-a---        2/5/2022   7:56 AM            519 enum_acls.ps1

PS C:\Users\sysadmin\Documents>

```

5. Test this script by moving to any directory (**cd C:\Windows**), and running **C:\Users\sysadmin\Documents\enum_acls.ps1** (enter the full path and file name).

You should see the ACL output of each file or subdirectory where you ran the script from.

```

Administrator: Windows PowerShell
PS C:\Users\sysadmin\Documents> cd C:\Windows\
PS C:\Windows> C:\Users\sysadmin\Documents\enum_acls.ps1
get-childitem : Access to the path 'C:\Windows\CSC' is denied.
At C:\Users\sysadmin\Documents\enum_acls.ps1:8 char:10
+ $items = get-childitem -path $directory -recurse
+
+ CategoryInfo          : PermissionDenied: (C:\Windows\CSC:String) [Get-ChildItem], UnauthorizedAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

get-childitem : Access to the path 'C:\Windows\System32\LogFiles\WMI\RtBackup' is denied.
At C:\Users\sysadmin\Documents\enum_acls.ps1:8 char:10
+ $items = get-childitem -path $directory -recurse
+
+ CategoryInfo          : PermissionDenied: (C:\Windows\System32\LogFiles\WMI\RtBackup:String) [Get-ChildItem], UnauthorizedAccessException
+ FullyQualifiedErrorId : DirUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

Directory: C:\Windows

Path                Owner             Access
----                ----             -----
addins              NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
appcompat           NT AUTHORITY\SYSTEM      NT SERVICE\TrustedInstaller Allow FullControl...
apppatch            NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
AppReadiness         NT AUTHORITY\SYSTEM      NT AUTHORITY\Authenticated Users Allow Read, Synchronize...
assembly            BUILTIN\Administrators    BUILTIN\Administrators Allow FullControl...
bcastdvr            NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
Boot                NT SERVICE\TrustedInstaller NT AUTHORITY\SYSTEM Allow -1610612736...
Branding            NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
CbsTemp              BUILTIN\Administrators    BUILTIN\Administrators Allow FullControl...
Containers           NT AUTHORITY\SYSTEM      NT SERVICE\TrustedInstaller Allow FullControl...
CSC                 NT AUTHORITY\SYSTEM      NT AUTHORITY\SYSTEM Allow FullControl
 Cursors             NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
debug               NT AUTHORITY\SYSTEM      APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Deny Fu...
diagnostics         NT SERVICE\TrustedInstaller NT AUTHORITY\SYSTEM Allow -1610612736...
DiagTrack            NT SERVICE\TrustedInstaller CREATOR OWNER Allow 268435456...
DigitalLocker        NT AUTHORITY\SYSTEM      NT SERVICE\TrustedInstaller Allow FullControl...

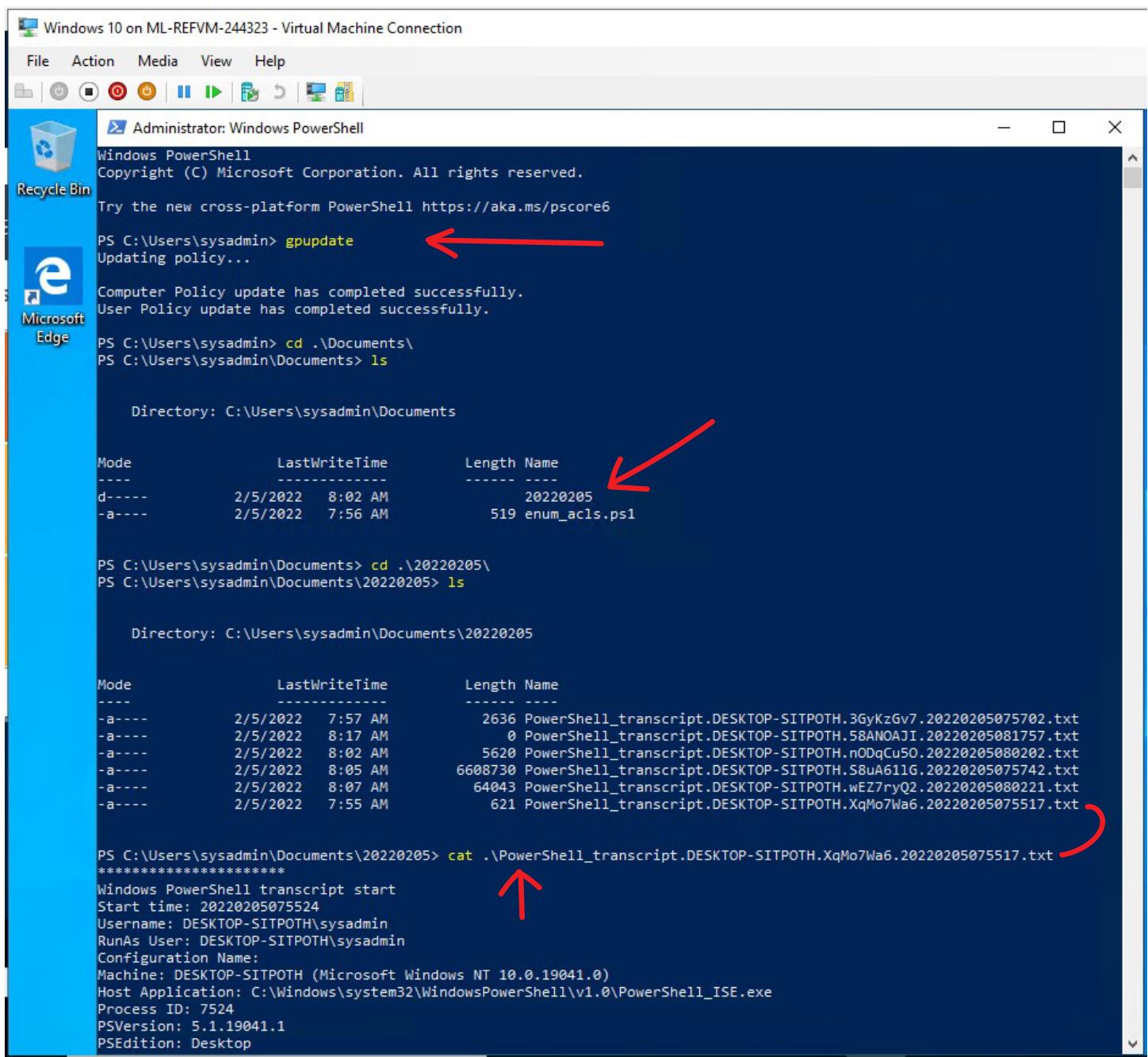
```

Bonus Task 5: Verify Your PowerShell Logging GPO

For this task we'll want to test and verify that our PowerShell logging GPO is working properly.

Instructions

- Ensure you're logged into the Windows 10 machine as **sysadmin | cybersecurity**.
 - Run **gpupdate** in an administrative PowerShell window to pull the latest Active Directory changes.
 - Close and relaunch PowerShell into an administrative session.
 - Navigate to a directory you want to see the ACLs in. You can go to C:\Windows, as you did in Task 4.
 - Run the **enum_acls.ps1** script using the full file path and name such as the one in Task 4.
 - Check the **C:\Users\sysadmin\Documents** for your new logs.
- *You should see a directory with the current date (for example, **20200908**) as the directory name. Your new transcribed PowerShell logs should be inside.



The screenshot shows a Windows 10 desktop with a pinned Microsoft Edge icon and a Recycle Bin icon. An Administrator: Windows PowerShell window is open. A red arrow points to the command `gpupdate`. Another red arrow points to the output of the `ls` command, which lists a file named `enum_acls.ps1` in the `20220205` directory. A third red arrow points to the transcript file `PowerShell_transcript.DESKTOP-SITPOTH.XqMo7Wa6.20220205075517.txt`.

```

Windows 10 on ML-REFVM-244323 - Virtual Machine Connection
File Action Media View Help
Recycle Bin
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\sysadmin> gpupdate
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.

PS C:\Users\sysadmin> cd ..\Documents\
PS C:\Users\sysadmin\Documents> ls

Directory: C:\Users\sysadmin\Documents

Mode LastWriteTime Length Name
---- -- -- -
d----- 2/5/2022 8:02 AM 20220205
-a---- 2/5/2022 7:56 AM 519 enum_acls.ps1

PS C:\Users\sysadmin\Documents> cd ..\20220205\
PS C:\Users\sysadmin\Documents\20220205> ls

Directory: C:\Users\sysadmin\Documents\20220205

Mode LastWriteTime Length Name
---- -- -- -
-a---- 2/5/2022 7:57 AM 2636 PowerShell_transcript.DESKTOP-SITPOTH.3GyKzGv7.20220205075702.txt
-a---- 2/5/2022 8:17 AM 0 PowerShell_transcript.DESKTOP-SITPOTH.58ANOAJI.20220205081757.txt
-a---- 2/5/2022 8:02 AM 5620 PowerShell_transcript.DESKTOP-SITPOTH.n0DqCu50.20220205080202.txt
-a---- 2/5/2022 8:05 AM 6608730 PowerShell_transcript.DESKTOP-SITPOTH.S8uA611G.20220205075742.txt
-a---- 2/5/2022 8:07 AM 64043 PowerShell_transcript.DESKTOP-SITPOTH.wEZ7ryQ2.20220205080221.txt
-a---- 2/5/2022 7:55 AM 621 PowerShell_transcript.DESKTOP-SITPOTH.XqMo7Wa6.20220205075517.txt

PS C:\Users\sysadmin\Documents\20220205> cat ..\PowerShell_transcript.DESKTOP-SITPOTH.XqMo7Wa6.20220205075517.txt
*****
Windows PowerShell transcript start
Start time: 20220205075524
Username: DESKTOP-SITPOTH\sysadmin
RunAs User: DESKTOP-SITPOTH\sysadmin
Configuration Name:
Machine: DESKTOP-SITPOTH (Microsoft Windows NT 10.0.19041.0)
Host Application: C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell_ISE.exe
Process ID: 7524
PSVersion: 5.1.19041.1
PSEdition: Desktop

```

Windows 10 on ML-REFVM-244323 - Virtual Machine Connection

File Action Media View Help

Recycle Bin Microsoft Edge

Administrator: Windows PowerShell

```
PS C:\Users\sysadmin\Documents\20220205> cat .\PowerShell_transcript.DESKTOP-SITPOTH.XqMo7Wa6.20220205075517.txt
*****
Windows PowerShell transcript start
Start time: 20220205075524
Username: DESKTOP-SITPOTH\sysadmin
RunAs User: DESKTOP-SITPOTH\sysadmin
Configuration Name:
Machine: DESKTOP-SITPOTH (Microsoft Windows NT 10.0.19041.0)
Host Application: C:\Windows\system32\WindowsPowerShell\v1.0\PowerShell_ISE.exe
Process ID: 7524
PSVersion: 5.1.19041.1
PSEdition: Desktop
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0, 5.1.19041.1
BuildVersion: 10.0.19041.1
CLRVersion: 4.0.30319.42000
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
*****
*****
Command start time: 20220205075524
*****
PS>[Microsoft.Windows.PowerShell.Gui.Internal.HostTextWriter]::RegisterHost($host.ui)
*****
Command start time: 20220205075525
*****
PS>filter more { $_ }
*****
Command start time: 20220205075525
*****
PS>
function psEdit([Parameter(Mandatory=$true)]$filenames)
{
    foreach ($filename in $filenames)
    {
        dir $filename | where {!$_.PSIsContainer} | %{
            $psISE.CurrentPowerShellTab.Files.Add($_.FullName) > $null
        }
    }
}
*****
Command start time: 20220205075525
*****
PS>$OutputEncoding = [System.Console]::OutputEncoding
*****
Command start time: 20220205075525
*****
PS>ipmo ISE
*****
Command start time: 20220205075526
*****
PS>CommandInvocation(Set-Variable): "Set-Variable"
>> ParameterBinding(Set-Variable): name="Name"; value="profile"
```