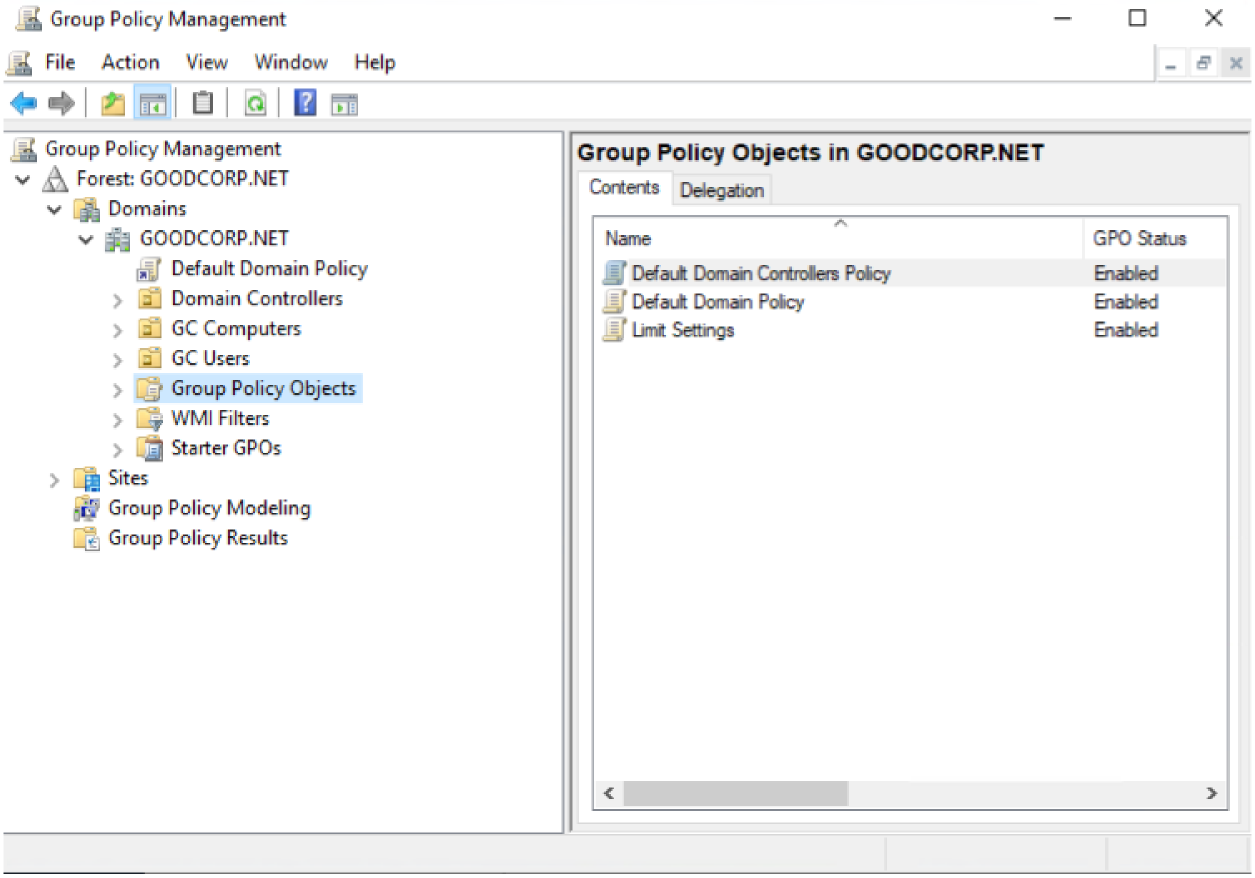
**For my own sake, I will be going through each of these step by step to show how I accomplished the deliverable**

**Deliverable for Task 1:** Take a screenshot of all the GPOs created for this homework assignment. To find these, launch the Group Policy Management tool, select **Group Policy Objects**, and take a screenshot of the GPOs you've created.

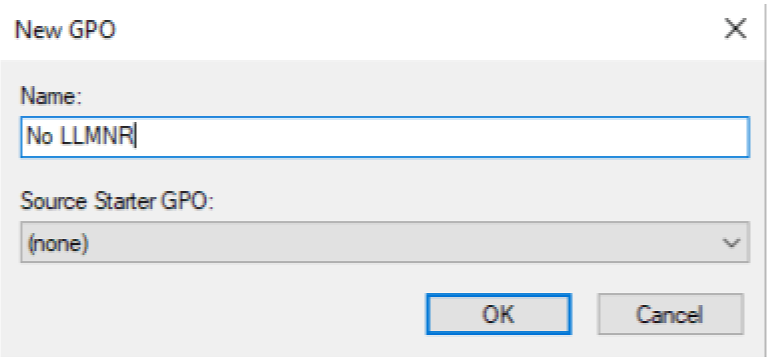
From the desktop, I searched "Group Policy Management" and opened the program.



Group Policy Objects in GOODCORP.NET

| Name | GPO Status |
|---|---|
| Default Domain Controllers Policy | Enabled |
| Default Domain Policy | Enabled |
| Limit Settings | Enabled |

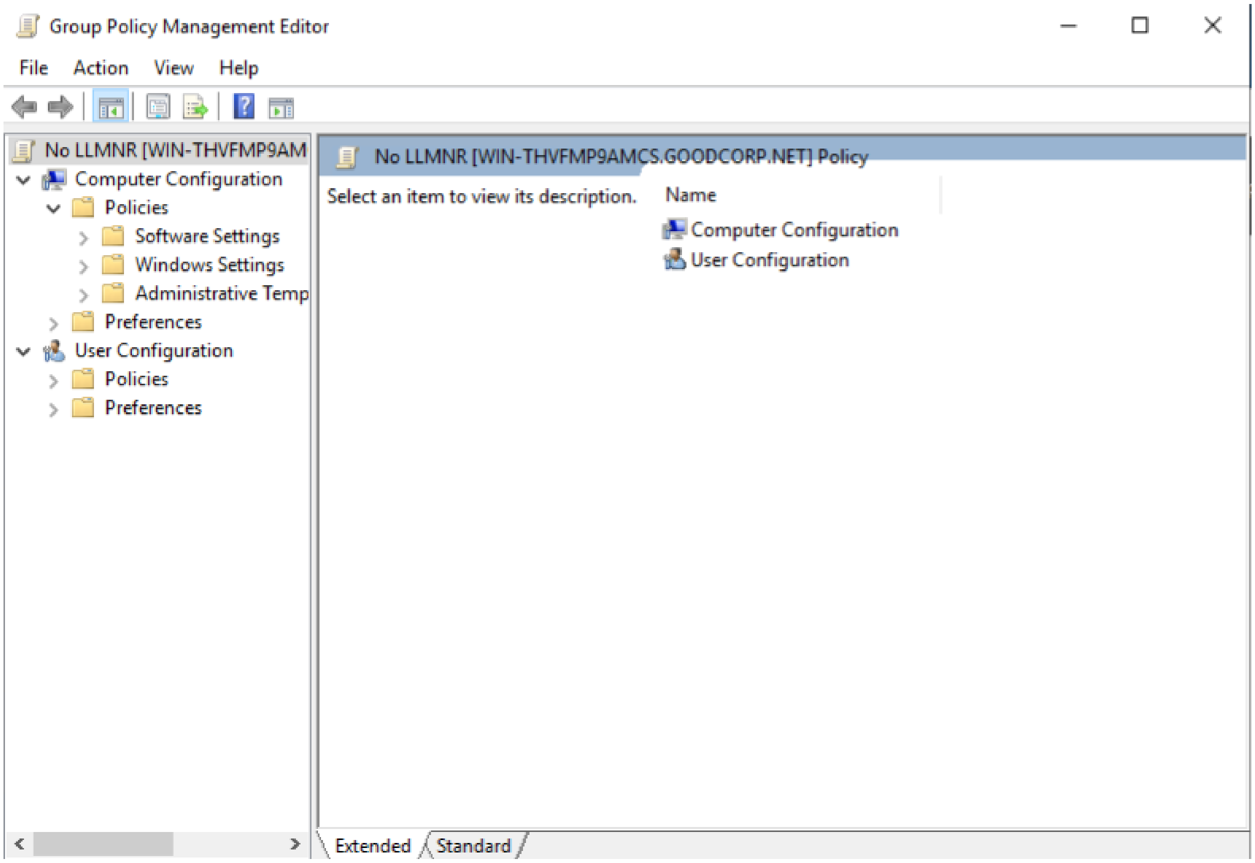
From there, we go to "Group Policy Objects" and create "new" and call it "No LLMNR"



Name: No LLMNR

Source Starter GPO: (none)

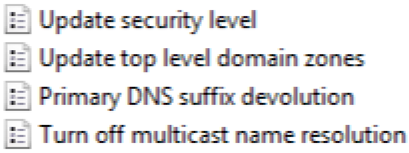
We then right click our new policy and go to “edit”, which brings us the following:



Select an item to view its description.

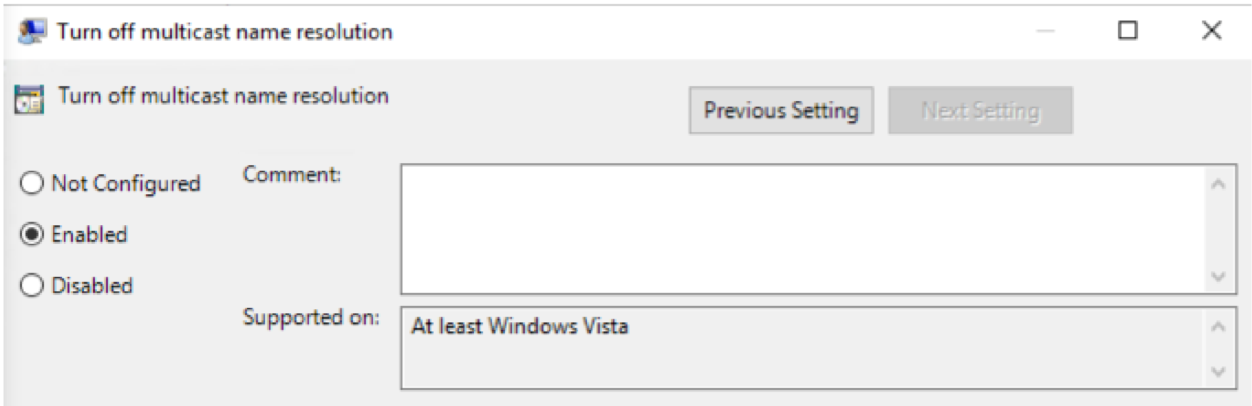
Once here, follow the directory: Computer Configuration\Policies\Administrative Templates\Network\DNS Client.

Now we look for “Turn off multicast name resolution”



Update security level
Update top level domain zones
Primary DNS suffix devolution
Turn off multicast name resolution

There it is at the bottom. Now to enable the policy.



Turn off multicast name resolution

Turn off multicast name resolution

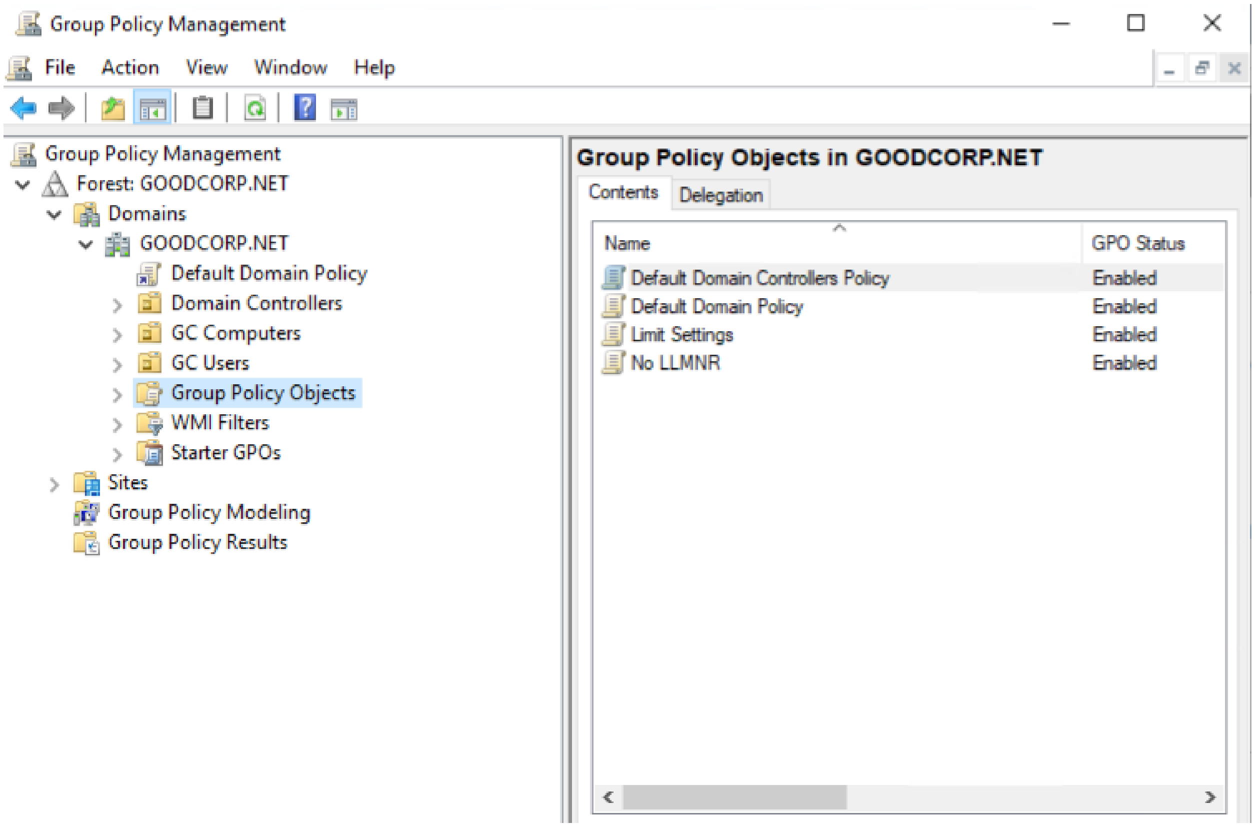
○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Vista

We click “apply” then “OK”

The last step is to return to the overall Group Policy Management. Right click on GC Computers, “Link an existing GPO” and select the GPO we created titled “No LLMNR” Displayed below is the final deliverable requested of all the GPO’s



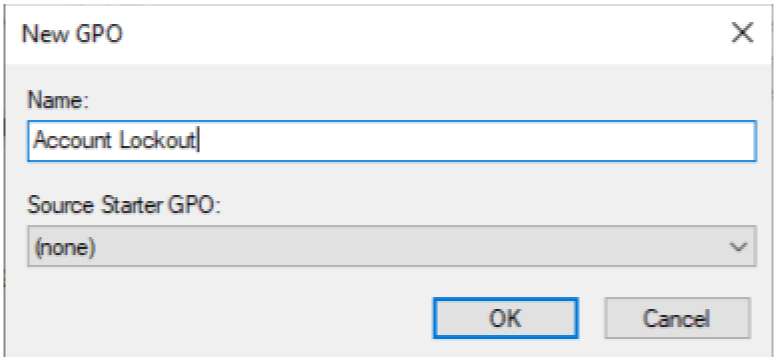
- Group Policy Management
  - Forest: GOODCORP.NET
    - Domains
      - GOODCORP.NET
        - Default Domain Policy
        - Domain Controllers
        - GC Computers
        - GC Users
        - Group Policy Objects
        - WMI Filters
        - Starter GPOs
    - Sites
    - Group Policy Modeling
    - Group Policy Results

Group Policy Objects in GOODCORP.NET

| Name | GPO Status |
|---|---|
| Default Domain Controllers Policy | Enabled |
| Default Domain Policy | Enabled |
| Limit Settings | Enabled |
| No LLMNR | Enabled |

**Deliverable for Task 2:** Submit a screenshot of the different Account Lockout policies in Group Policy Management Editor. It should show the three values you set under the Policy and Policy Setting columns.

Same as above, go to “Group Policy Objects” and create “new” and call it “Account Lockout”

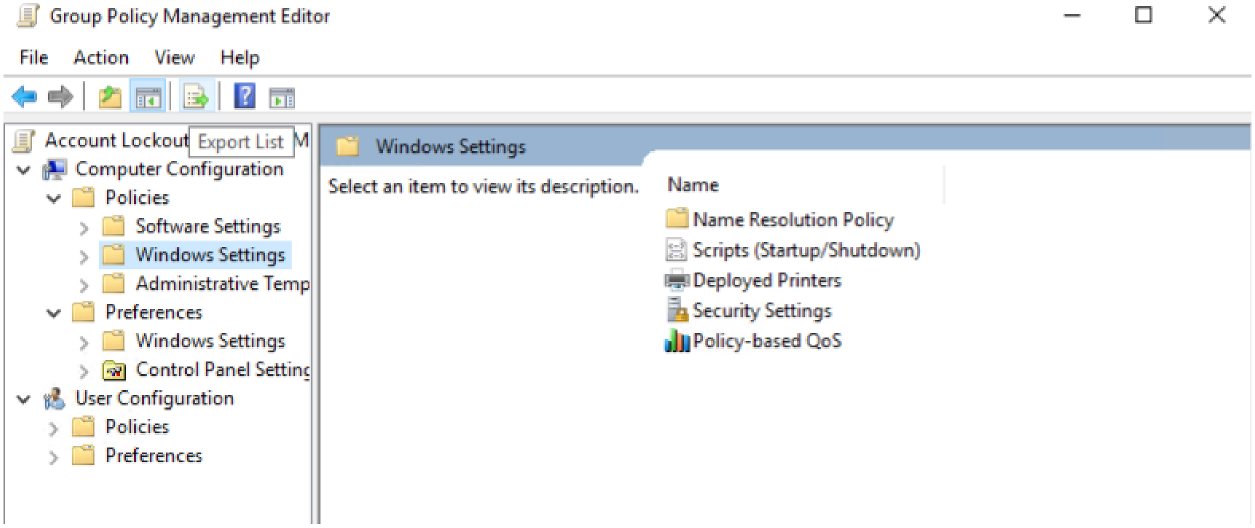


New GPO

Name: Account Lockout

Source Starter GPO: (none)

So now we’re going to fiddle with some security settings. In the GPO editor. We’re going to follow: Computer Configuration/Windows Settings/Security Settings



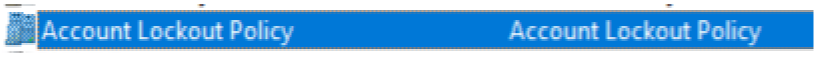
Group Policy Management Editor

Select an item to view its description.

Open up security settings. We’re trying to edit account lockouts, so the only relevant item in here is “Account Policies” which has “account lockout policies” in the description. So let’s go there.

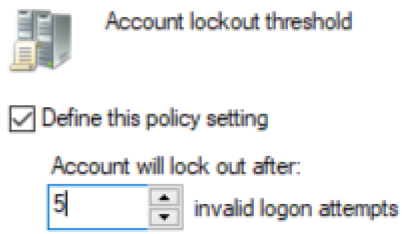
Account Policies    Password and account lockout policies

Then here



Account Lockout Policy    Account Lockout Policy

Lockout Threshold, or how many attempts the user has to try the password, will be set to 5 attempts.



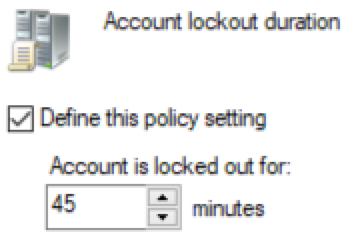
Account lockout threshold

☑ Define this policy setting

Account will lock out after:

5 invalid logon attempts

Account lockout duration, or how long the user will be locked out after 5 incorrect attempts, will be set at 45 minutes.



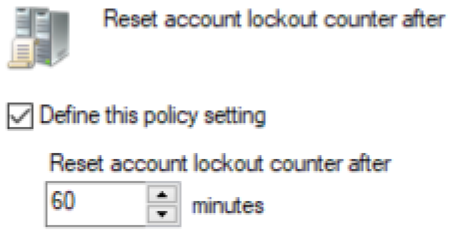
Account lockout duration

☑ Define this policy setting

Account is locked out for:

45 minutes

Finally, the attempts can be reset within, let's say 60 minutes.



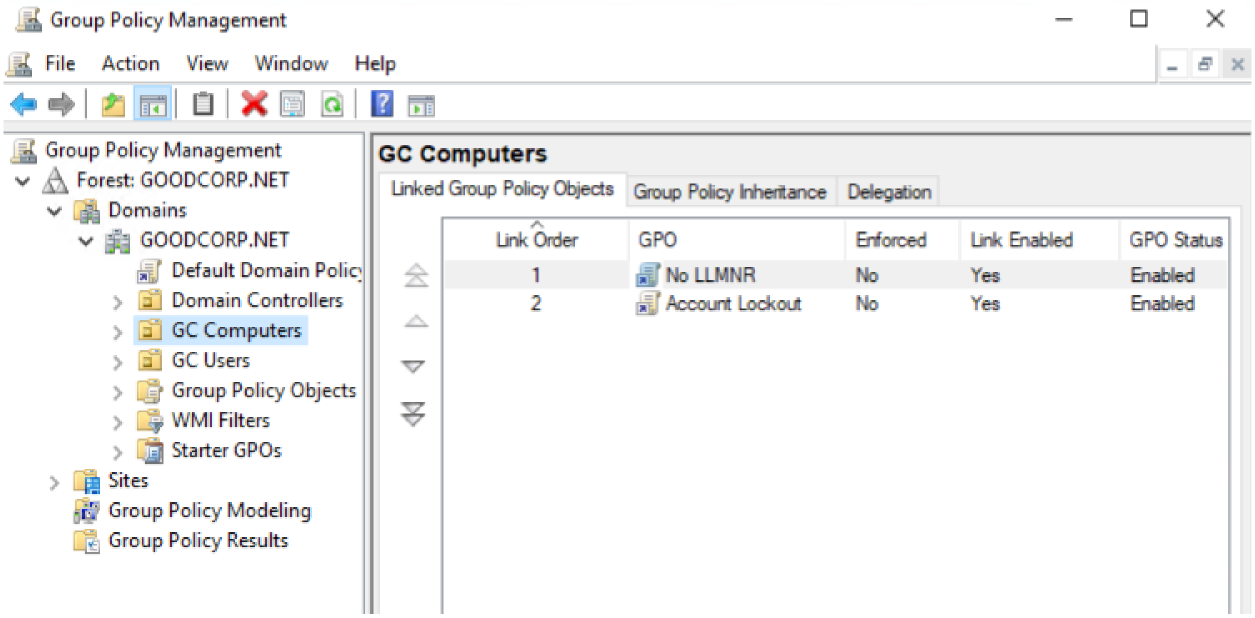
Reset account lockout counter after

☑ Define this policy setting

Reset account lockout counter after

60 minutes

Now back to the Group Policy Manager. Like the last setting, we will link this GPO to the "GC Computers"

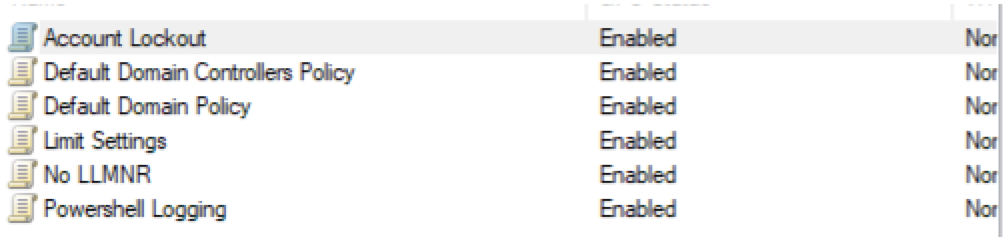


| Link Order | GPO | Enforced | Link Enabled | GPO Status |
|---|---|---|---|---|
| 1 | No LLMNR | No | Yes | Enabled |
| 2 | Account Lockout | No | Yes | Enabled |

And there there it is.

**Deliverable for Task 3:** Submit a screenshot of the different Windows PowerShell policies within the Group Policy Management Editor. Four of these should be enabled.

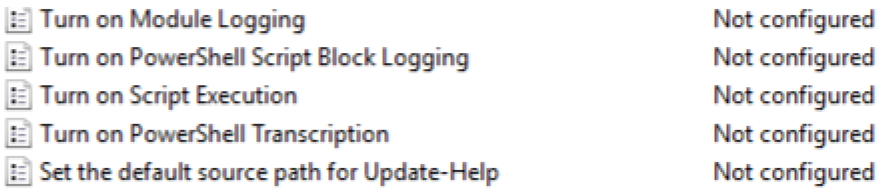
Same as the previous two, we're going to create a GPO called "Powershell Logging"



| | | |
|---|---|---|
| Account Lockout | Enabled | Nor |
| Default Domain Controllers Policy | Enabled | Nor |
| Default Domain Policy | Enabled | Nor |
| Limit Settings | Enabled | Nor |
| No LLMNR | Enabled | Nor |
| Powershell Logging | Enabled | Nor |

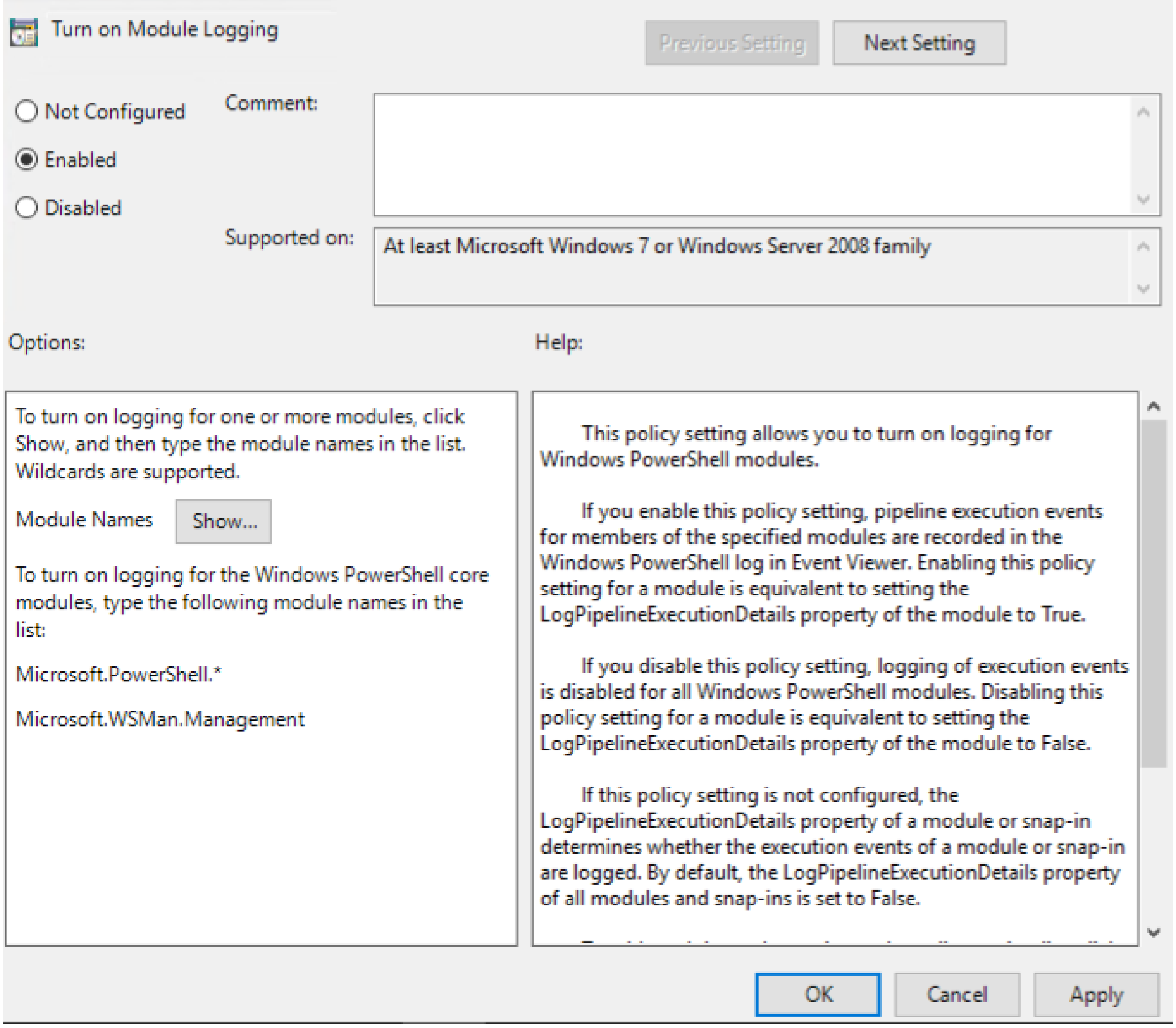
There it is at the bottom. Next we're going to edit the newly created GPO. We have to find Powershell specific settings, so we'll follow the directory of Computer Configuration/Administrative Templates/Windows Components/Windows Powershell

Once you find it, you will see these settings:



| | |
|---|---|
| Turn on Module Logging | Not configured |
| Turn on PowerShell Script Block Logging | Not configured |
| Turn on Script Execution | Not configured |
| Turn on PowerShell Transcription | Not configured |
| Set the default source path for Update-Help | Not configured |

We want to enable  "Turn on Module Logging"



Turn on Module Logging

○ Not Configured

● Enabled

○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

To turn on logging for one or more modules, click Show, and then type the module names in the list. Wildcards are supported.

Module Names

To turn on logging for the Windows PowerShell core modules, type the following module names in the list:

Microsoft.PowerShell.*

Microsoft.WSMan.Management

Help:

This policy setting allows you to turn on logging for Windows PowerShell modules.

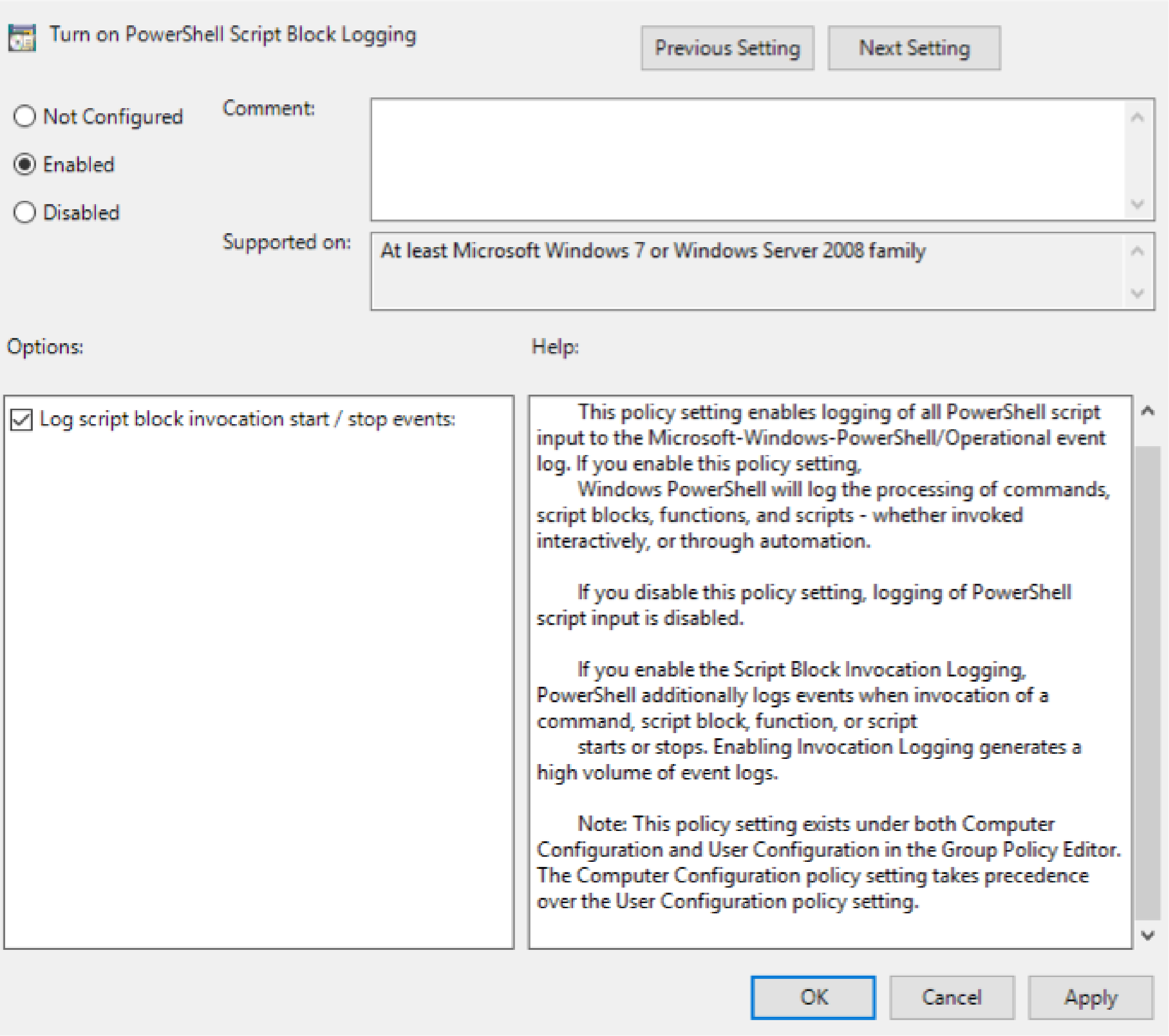
If you enable this policy setting, pipeline execution events for members of the specified modules are recorded in the Windows PowerShell log in Event Viewer. Enabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to True.

If you disable this policy setting, logging of execution events is disabled for all Windows PowerShell modules. Disabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to False.

If this policy setting is not configured, the LogPipelineExecutionDetails property of a module or snap-in determines whether the execution events of a module or snap-in are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

Before we hit “apply” we also need to log all powershell modules. For these we can use the wildcard (*). Click “show” and create a value, type in “*” to signify all. Click “ok”, then “apply” on the main module editing, then “ok” again. It will now be enabled.

Next we enable “PowerShell Script Block Logging”



○ Not Configured
● Enabled
○ Disabled

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

☑ Log script block invocation start / stop events:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting,
Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

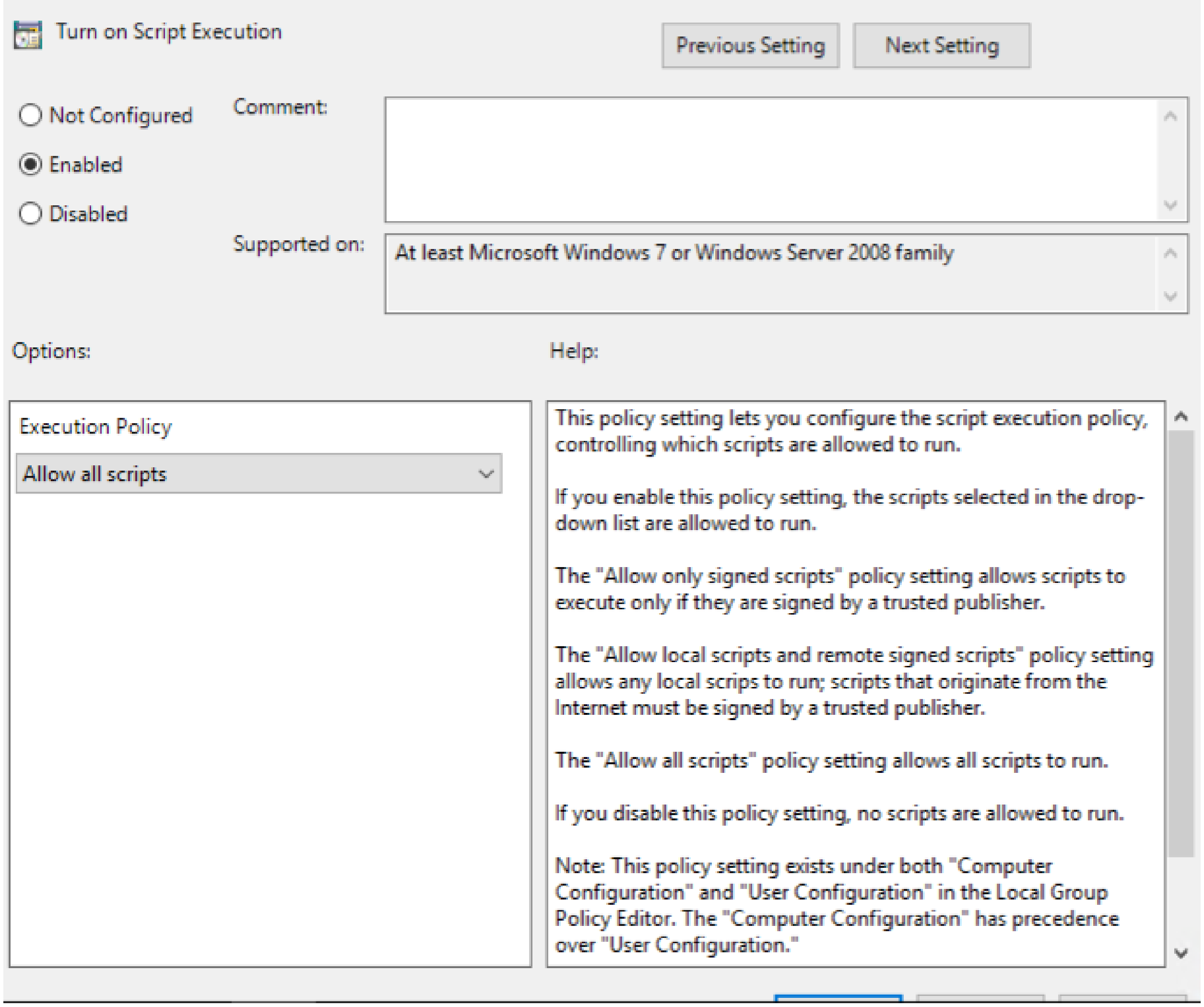
If you disable this policy setting, logging of PowerShell script input is disabled.

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script
starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Enabled, and the only option has been enabled as well. Hit “apply” then “ok”

Script Execution is next:



Turn on Script Execution

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

Execution Policy: Allow all scripts

Help:

This policy setting lets you configure the script execution policy, controlling which scripts are allowed to run.

If you enable this policy setting, the scripts selected in the drop-down list are allowed to run.

The "Allow only signed scripts" policy setting allows scripts to execute only if they are signed by a trusted publisher.

The "Allow local scripts and remote signed scripts" policy setting allows any local scrips to run; scripts that originate from the Internet must be signed by a trusted publisher.

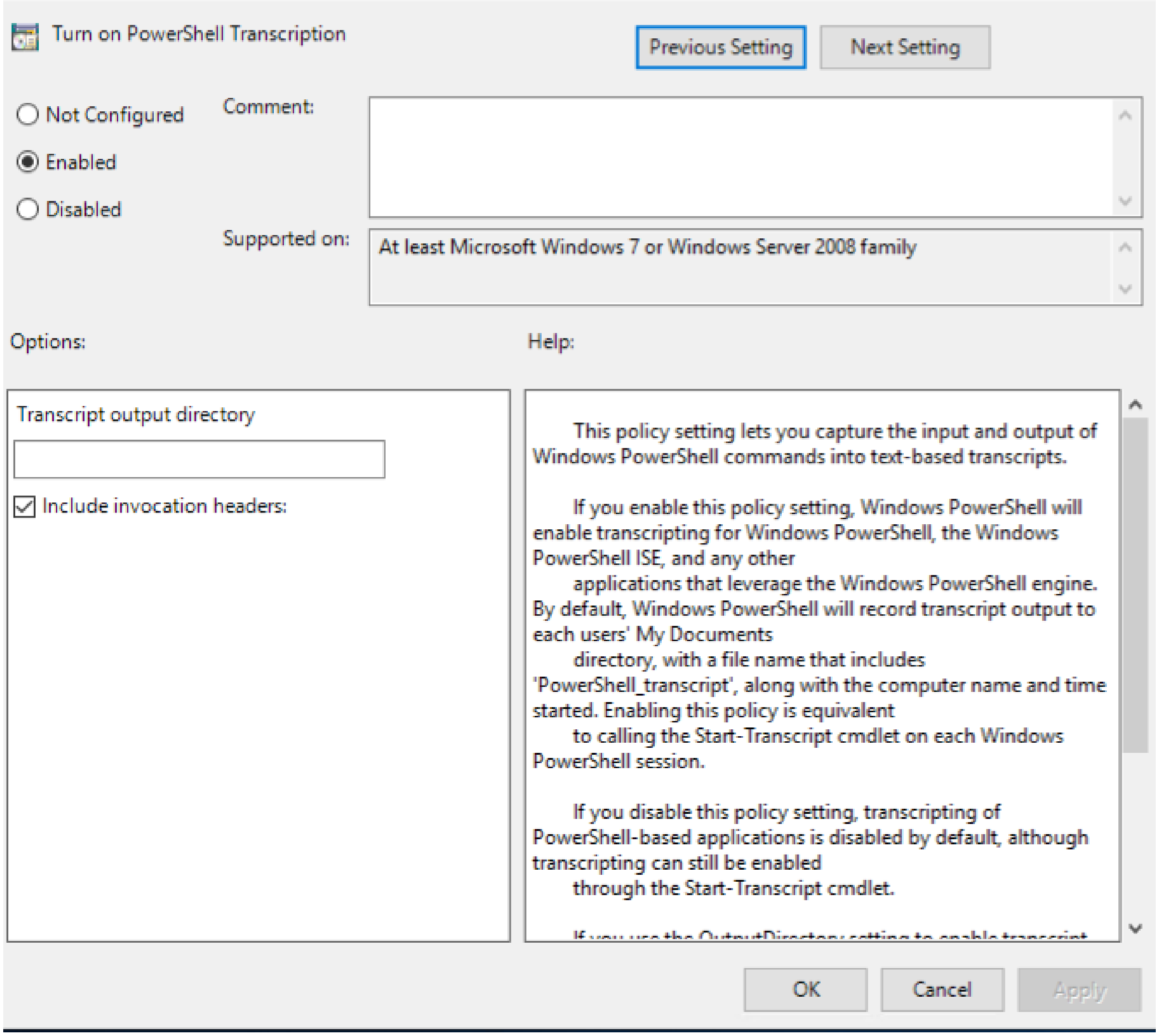
The "Allow all scripts" policy setting allows all scripts to run.

If you disable this policy setting, no scripts are allowed to run.

Note: This policy setting exists under both "Computer Configuration" and "User Configuration" in the Local Group Policy Editor. The "Computer Configuration" has precedence over "User Configuration."

Enabled and set the Execution Policy to “Allow all scripts”. Hit “Apply” then “ok”

Lastly, we enable Powershell Transcription:



○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

Transcript output directory

☑ Include invocation headers:

Help:

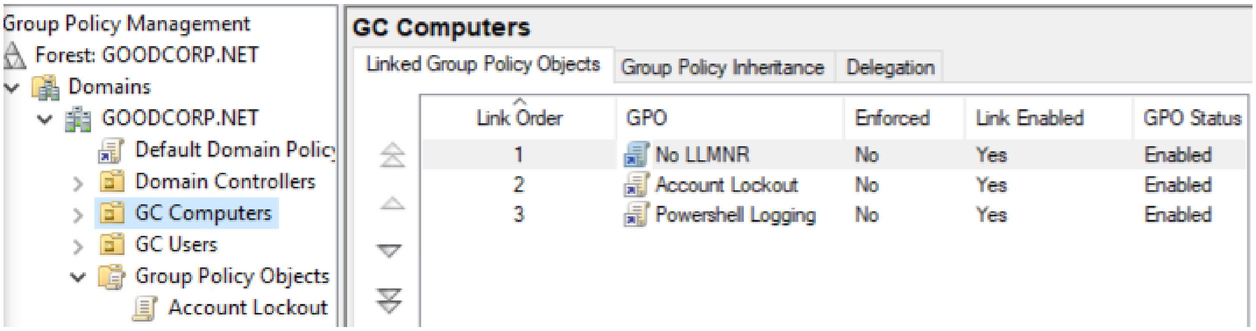
This policy setting lets you capture the input and output of Windows PowerShell commands into text-based transcripts.

If you enable this policy setting, Windows PowerShell will enable transcripting for Windows PowerShell, the Windows PowerShell ISE, and any other applications that leverage the Windows PowerShell engine. By default, Windows PowerShell will record transcript output to each users' My Documents directory, with a file name that includes 'PowerShell_transcript', along with the computer name and time started. Enabling this policy is equivalent to calling the Start-Transcript cmdlet on each Windows PowerShell session.

If you disable this policy setting, transcripting of PowerShell-based applications is disabled by default, although transcripting can still be enabled through the Start-Transcript cmdlet.

If you use the OutputDirectory setting to enable transcript

With everything necessary now enabled, we have to link it to GC Computers.



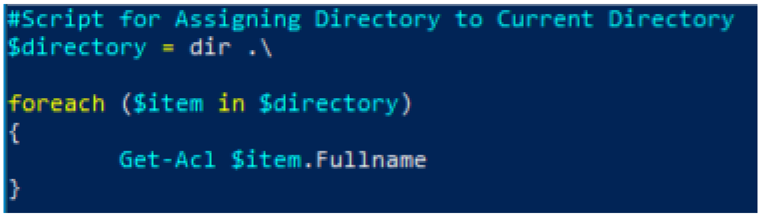
| Link Order | GPO | Enforced | Link Enabled | GPO Status |
|---|---|---|---|---|
| 1 | No LLMNR | No | Yes | Enabled |
| 2 | Account Lockout | No | Yes | Enabled |
| 3 | Powershell Logging | No | Yes | Enabled |

There it is. Now onto Task 4.

**Deliverable for Task 4:** Submit a copy of your enum_acls.ps1 script.

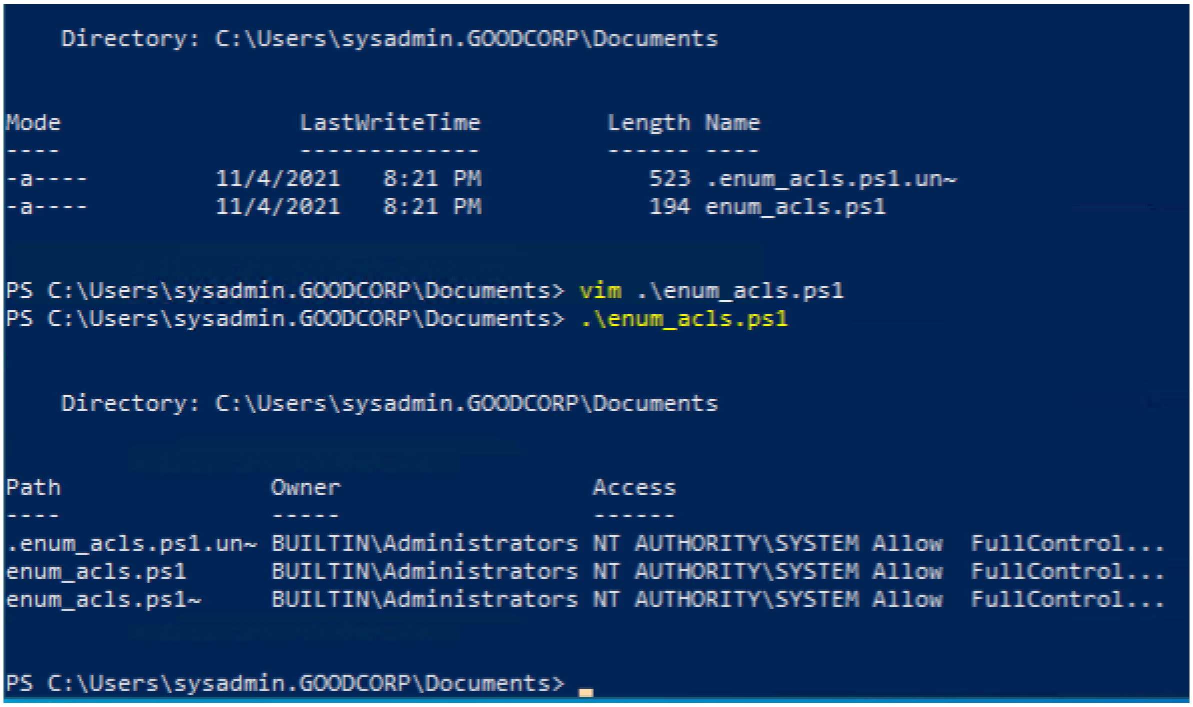
First, let’s navigate to documents in Powershell since that’s where our finished script will go. You can do this with cd .\Documents\.

Once in here, we can start making our script.



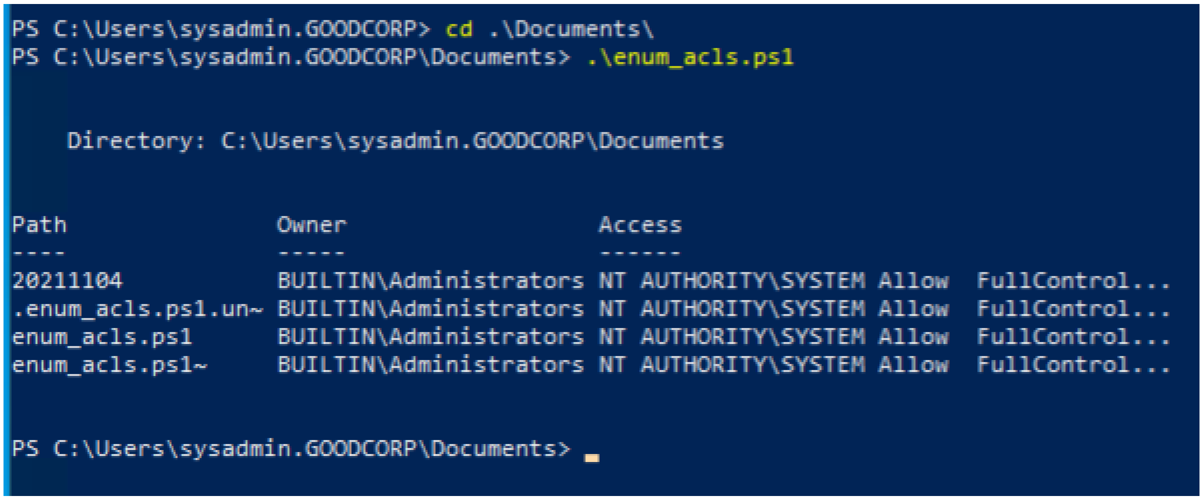
```
#Script for Assigning Directory to Current Directory
$directory = dir .\

foreach ($item in $directory)
{
        Get-Acl $item.Fullname
}
```

There’s our script. Now to test it.



```
    Directory: C:\Users\sysadmin.GOODCORP\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         11/4/2021   8:21 PM            523 .enum_acls.ps1.un~
-a----         11/4/2021   8:21 PM            194 enum_acls.ps1


PS C:\Users\sysadmin.GOODCORP\Documents> vim .\enum_acls.ps1
PS C:\Users\sysadmin.GOODCORP\Documents> .\enum_acls.ps1


    Directory: C:\Users\sysadmin.GOODCORP\Documents


Path                Owner                  Access
----                -----                  ------
.enum_acls.ps1.un~ BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...
enum_acls.ps1      BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...
enum_acls.ps1~     BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...


PS C:\Users\sysadmin.GOODCORP\Documents> _
```

**Deliverable for Bonus Task 5:** Submit a screenshot of the contents of one of your transcribed PowerShell logs or a copy of one of the logs.



```
PS C:\Users\sysadmin.GOODCORP> cd .\Documents\
PS C:\Users\sysadmin.GOODCORP\Documents> .\enum_acls.ps1


    Directory: C:\Users\sysadmin.GOODCORP\Documents


Path               Owner                  Access
----               -----                  ------
20211104           BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...
.enum_acls.ps1.un~ BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...
enum_acls.ps1      BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...
enum_acls.ps1~     BUILTIN\Administrators NT AUTHORITY\SYSTEM Allow  FullControl...


PS C:\Users\sysadmin.GOODCORP\Documents> _
```