In this PowerShell script, we aim to prevent the execution of sethc.exe on a list of remote computers provided in a CSV file. I'll explain each part of the code in detail, assuming you're new to PowerShell.

Text

Description automatically generated

* $csvFilePath = ".\computers.csv":
  + This line sets a variable called $csvFilePath to the relative path of the CSV file containing the list of computer names.
  + The [.] represents the current directory, and the file is named computers.csv.

Text

Description automatically generated

* $computers = Import-Csv -Path $csvFilePath:
  + This line imports the contents of the CSV file specified by $csvFilePath using the Import-Csv cmdlet.
  + The contents of the CSV file are stored in the variable $computers.



* foreach ($computer in $computers):
  + This line starts a foreach loop.
  + It iterates through each item in the $computers collection.
  + Each item (i.e., each computer) will be temporarily stored in the $computer variable.

A screenshot of a computer

Description automatically generated with medium confidence

* $computerName = $computer.ComputerName:
  + This line extracts the ComputerName property from the current $computer item and stores it in the $computerName variable.



* try { ... } catch { ... }:
  + This block is used for error handling.
  + If an error occurs within the try block, execution will jump to the catch block, allowing us to handle the error gracefully.

Text

Description automatically generated

* $session = New-PSSession -ComputerName $computerName:
  + Inside the try block, this line attempts to create a new remote PowerShell session with the target computer using the New-PSSession cmdlet.
  + If successful, the session is stored in the $session variable.
  + If an error occurs, the catch block is executed, and a message will be displayed.
  + The continue statement will then skip the rest of the loop iteration, moving on to the next computer in the list.



* Invoke-Command -Session $session -ScriptBlock { ... }:
  + This line uses the Invoke-Command cmdlet to execute a script block (the code within the curly braces { ... }) on the remote computer within the established PowerShell session ($session).

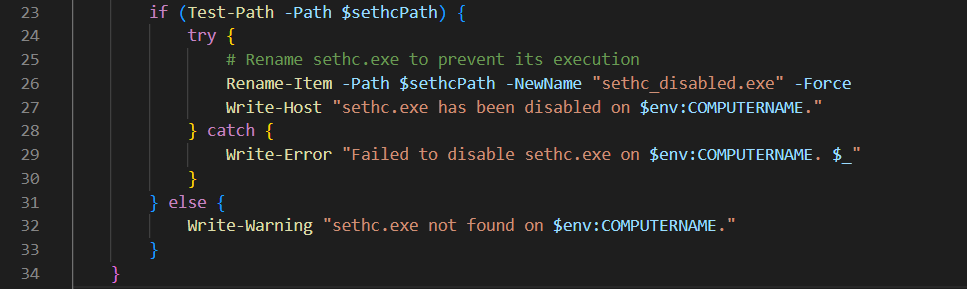
Text

Description automatically generated

* $sethcPath = "$env:SystemRoot\System32\sethc.exe":
  + Inside the script block, this line sets the $sethcPath variable to the full path of sethc.exe on the remote computer.
  + The $env:SystemRoot part is an environment variable that points to the Windows directory (usually C:\Windows).



* if (Test-Path -Path $sethcPath) { ... } else { ... }:
  + This line tests if the sethc.exe file exists at the specified path using the Test-Path cmdlet.
  + If the file exists, the code within the if block will be executed.
  + If not, the code within the else block will be executed.



* Rename-Item -Path $sethcPath -NewName "sethc\_disabled.exe" -Force:
  + Inside the if block, this line attempts to rename the sethc.exe file to sethc\_disabled.exe using the Rename-Item cmdlet.
  + The -Force flag ensures that the renaming happens even if the file is read-only.



* Write-Host, Write-Error, and Write-Warning:
  + These cmdlets are used to display informational, error, or warning messages, respectively.
  + For example, Write-Host is used to display a success message when sethc.exe is disabled.
  + Write-Error is used to display an error message when an error occurs, and Write-Warning is used to display a warning message when sethc.exe is not found on the remote computer.

Text

Description automatically generated

* Remove-PSSession -Session $session:
  + After executing the commands on the remote computer, this line closes the remote PowerShell session using the Remove-PSSession cmdlet.
  + It's important to close sessions when they are no longer needed to free up system resources.

Graphical user interface, text

Description automatically generated

In summary, this PowerShell script reads a list of computer names from a CSV file, establishes remote PowerShell sessions with each computer, checks if the sethc.exe file exists on the remote computers, and if it does, renames it to prevent execution. The script also handles errors and provides feedback using Write-Host, Write-Error, and Write-Warning.