1. Remote Control

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001 |
| More information | Introduction to remote control in System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/core/clients/manage/remote-control/introduction-to-remote-control>  Configuring remote control in System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/core/clients/manage/remote-control/configuring-remote-control>  How to audit remote control usage in System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/core/clients/manage/remote-control/audit-remote-control-usage> |
| Description | In this chapter, we will be creating a Device Client Setting to be used for Remote Tools as well as configuring, validating and monitoring access via SCCM Remote Control |

* 1. Creating Device Settings for Remote Tools

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select Client Settings and click Create Custom Client Device Settings |
| 04. Type Remote Control for Windows 10 on Name and select Remote Tools under Select the custom settings to be enforced on client devices |
| 05. Click Remote Tools |
| 06. Click Configure and select Enable Remote Control on client computers and Domain  Note: Enabling Private and Public should only be used for non-domain machines. Using this option for domain machines may decrease its security depending on the settings you have selected. As example, if you select the Remote Desktop, the connection can be initiated from any network. |
| 07. Under Remote Tools click Set Viewers and add CLASSROOM\SCCM Remote Tools. Click Ok  Note: This option is only required if the operator does not have administrative rights on target machines or if the Grant Remote Control permission to local Administrators group is set to No |
| 08. Under remote tools perform the following changes:   * Prompt user for Remote Control permission: yes * Play a sound on client: None * Manage Remote Desktop settings: Yes * Allow Permitted viewers to connect by using Remote Desktop Connection: Yes * Require network level authentication on computers that run Windows Vista operating system and later versions: No   Note: These settings are being used in a lab environment where security should not be an issue.  Click Ok |
| 09. Select the Remote Control for Windows 10 and click Deploy |
| 10. Under select collection, click Windows 10 Workstations and click Ok |
| 11. Select Deployments and confirm that the Client Settings has been deployed to the collection |

This can also be achieved via PowerShell using the commands below:

$ClientSettingsName = "Remote Control for Windows 10"

New-CMClientSetting -Name "$ClientSettingsName" -Type Device

Set-CMClientSetting -Name "$ClientSettingsName" -RemoteToolsSettings -AccessLevel FullControl -AllowPermittedViewersToRemoteDesktop $True -AllowRemoteControlOfUnattendedComputer $True -AudibleSignal PlayNoSound -FirewallExceptionProfile Domain -ManageRemoteDesktopSetting $True -ManageSolicitedRemoteAssistance $True -ManageUnsolicitedRemoteAssistance $True -PermittedViewer "CLASSROOM\SCCM Remote Tools" -RemoteAssistanceAccessLevel FullControl -RequireAuthentication $False -PromptUserForPermission $True

Start-CMClientSettingDeployment -ClientSettingName "$ClientSettingsName" -CollectionName "Windows 10 Workstations"

* 1. Checking the Policies that should be applied

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Select a Windows 10 Client and click Client Settings->Resultant Client Settings |
| 04. Once the Resultant Client Settings open, check the Remote Tools and confirm the changes have been made. Click Ok |

This can also be achieved via PowerShell using the commands below:

$DeviceList = get-cmdevice -Fast | where-object {$\_.DeviceOS -like 'Microsoft Windows NT Workstation 10.0\*'}

Get-CMResultantSettings -Name ($DeviceList[0].Name) -SettingsType Device

* 1. Validating Remote Control

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the SCCM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Open Computer Management |
| 05. Expand Local users and Groups and select Groups |
| 06. Open properties of the ConfigMgr Remote Control Users and confirm that the Permitted viewers have been added to this group |

This can also be achieved via PowerShell using the commands below:

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$groupName = "ConfigMgr Remote Control Users"

$LocalGroup = [ADSI]("WinNT://./$groupName,group")

$GMembers = $LocalGroup.psbase.invoke("Members")

$gmembers | foreach { $\_.GetType().InvokeMember("Name",'GetProperty', $null, $\_, $null) }

* 1. Starting Remote Control

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Assets and Compliance. |
| 02. Click Devices |
| 03. Select the Machine you want to remote access and click Start -> Remote Control |
| 04. The Remote Control will connect to the client |
| 05. On the client, it will need approve or Deny the connection if the client settings -> remote tools -> Prompt user for remote control permissions is set to yes |
| 06. Once the connection is established, the remote user will be able to access the machine |
| 07. The connection will remain active even when the user logs off.  Note: When using Enhanced session is being used on the virtual machine, once the users logs off, the session will be closed  Note: Remote Control to a blocked machine has changed from SCCM 2007 where it was not possible |

This can also be achieved via PowerShell using the commands below:

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath += "\CmRcViewer.exe"

$device = Get-CMDevice -Name "WKS0001"

if ($Device.IsClient -eq $true) { Start-Process -Filepath ("$ModulePath") -ArgumentList ("$($device.Name) \\$($servername)") } else { "Computer is not a SCCM Client" }

* 1. Monitoring Remote Access from Client

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. On the client, you can also review the following client logs:   * C:\Windows\ccm\Logs\cmRcService.log: Records information for the remote control service. |

This can also be achieved via PowerShell using the commands below:

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("c:\Windows\ccm\Logs\cmRcService.log")

* 1. Monitoring Remote Access from Server

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting and Click Reports |
| 03. Search for remote, select Remote Control – All remote control information and click Run |
| 04. Once the report opens, all information about successfully connections will be presented |
| 05. On the monitoring workspace, expand System Status and click Status Message Queries |
| 06. Search for Remote |
| 07. Select Remote Control Activity Targeted at a Specific System and click Show Messages |
| 08. Under Remote Control Activity Targeted at a Specific System select:   * Machine Name: WKS0001 * Time: Select date and time: 6 hours ago   Click Ok |
| 09. Verify the existence of Message ID 30076 |
| 10. Double click any 30076 messages to see its details. Once done, click Ok |
| 11. Verify the existence of Message ID 30077 |
| 12. Double click any 30077 messages to see its details. Once done, click Ok |

This can also be achieved via PowerShell using the commands below:

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

Invoke-CMReport -ReportPath "Status Messages - Audit/Remote Control - All remote control information" -SiteCode "$SiteCode" -SrsServerName "$servername"

#Query:

$Date = (Get-Date).AddHours(-6)

gwmi -Namespace "root\sms\site\_$SiteCode" -ComputerName "$servername" -query "select stat.\*, ins.\*, att1.\*, stat.Time from SMS\_StatusMessage as stat left join SMS\_StatMsgInsStrings as ins on stat.RecordID = ins.RecordID left join SMS\_StatMsgAttributes as att1 on stat.RecordID = att1.RecordID inner join SMS\_StatMsgInsStrings as ins2 on stat.RecordID = ins2.RecordID where stat.MessageType = 768 and stat.MessageID >= 30069 and stat.MessageID <= 30087 and ins2.InsStrIndex = 2 and ins2.InsStrValue = 'WKS0001' and stat.Time >= '$($Date.ToString('yyyy/MM/dd HH:mm:ss.fff'))' order by stat.Time desc"

$remoteaccesslist = gwmi -Namespace "root\sms\site\_$SiteCode" -ComputerName "$servername" -query "select stat.Time, stat.MessageID, ins.InsStrIndex, ins.InsStrValue, att1.AttributeID, att1.AttributeTime, att1.AttributeValue from SMS\_StatusMessage as stat left join SMS\_StatMsgInsStrings as ins on stat.RecordID = ins.RecordID left join SMS\_StatMsgAttributes as att1 on stat.RecordID = att1.RecordID inner join SMS\_StatMsgInsStrings as ins2 on stat.RecordID = ins2.RecordID where stat.MessageType = 768 and stat.MessageID >= 30069 and stat.MessageID <= 30087 and ins2.InsStrIndex = 2 and ins2.InsStrValue = 'WKS0001' and stat.Time >= '$($Date.ToString('yyyy/MM/dd HH:mm:ss.fff'))' order by stat.Time desc"

foreach ($remoteaccess in $remoteaccesslist) {

$props = @{ 'Time'=$remoteaccess.stat.Time;

'MessageID'=$remoteaccess.stat.MessageID

'InsStrIndex'=$remoteaccess.ins.InsStrIndex

'InsStrValue'=$remoteaccess.ins.InsStrValue

'AttributeID'=$remoteaccess.att1.AttributeID

'AttributeTime'=$remoteaccess.att1.AttributeTime

}

$obj = new-object -TypeName psobject -Property $props

write-output $obj

}