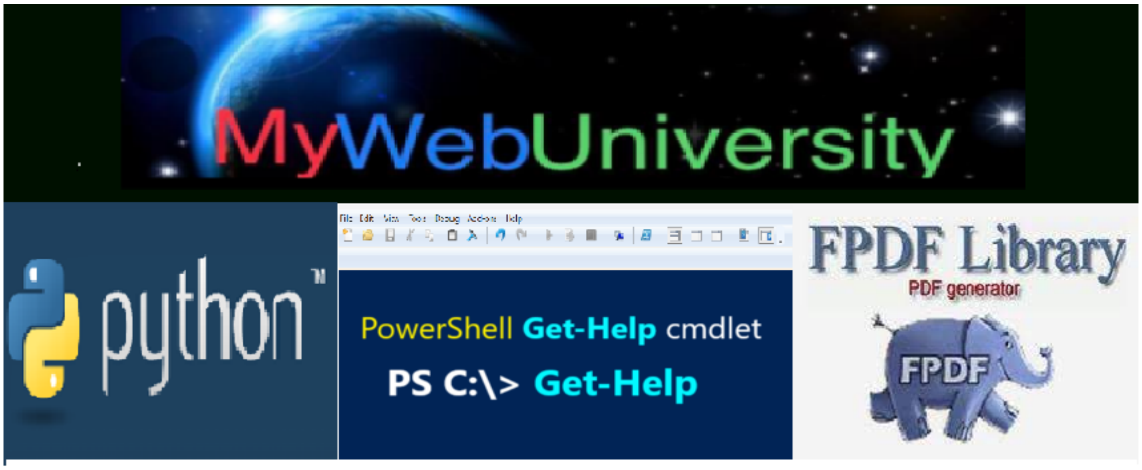

Windows PowerShell Get-Help on Cmdlet 'Disable-PSRemoting'

PS:\>Get-HELP Disable-PSRemoting -Full

NAME

Disable-PSRemoting

SYNOPSIS

Prevents PowerShell endpoints from receiving remote connections.

SYNTAX

Disable-PSRemoting [-Force] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Disable-PSRemoting` cmdlet blocks remote access to all Windows PowerShell session endpoint configurations on the local computer. This includes any endpoints created by PowerShell 6 or higher.

To re-enable remote access to all session configurations, use the `Enable-PSRemoting` cmdlet. This includes any endpoints created by PowerShell 6 or higher. To enable remote access to selected session configurations, use the AccessMode parameter of the `Set-PSSessionConfiguration` cmdlet. You can also use the

`Enable-PSSessionConfiguration` and `Disable-PSSessionConfiguration` cmdlets to enable and disable session configurations for all users. For more information about

session configurations, see about_Session_Configurations (About/about_Session_Configurations.md).

> [!NOTE] > Even after running `Disable-PSRemoting` you can still make loopback connections on the local > machine. A loopback connection is a PowerShell remote

session that originates from and connects to > the same local machine. Remote sessions from external sources remain blocked. For loopback > connections you must use

implicit credentials along the EnableNetworkAccess parameter. For > more information about loopback connections, see New-PSSession (New-PSSession.md).

To run this cmdlet, start Windows PowerShell with the Run as administrator option.

PARAMETERS

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required?   false

Position?   named

Default value   False

Accept pipeline input?   False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

None

You can't pipe objects to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

- Disabling the session configurations does not undo all the changes that were made by the `Enable-PSRemoting` or `Enable-PSSessionConfiguration` cmdlets. You

might have to undo the following changes manually.

1. Stop and disable the WinRM service.   2. Delete the listener that accepts requests on any IP address.   3. Disable

the firewall exceptions for WS-Management communications. 4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to members of the Administrators group on the computer.

A session configuration is a group of settings that define the environment for a session. Every session that connects to the computer must use one of the session configurations that are registered on the computer. By denying remote access to all session configurations, you effectively prevent remote users from establishing sessions that connect to the computer.

In Windows PowerShell 2.0, `Disable-PSRemoting` adds a Deny_All entry to the security descriptors of all session configurations. This setting prevents all users from creating user-managed sessions to the local computer. In Windows PowerShell 3.0, `Disable-PSRemoting` adds a Network_Deny_All entry to the security descriptors of all session configurations. This setting prevents users on other computers from creating user-managed sessions on the local computer, but allows users of the local computer to create user-managed loopback sessions.

In Windows PowerShell 2.0, `Disable-PSRemoting` is the equivalent of `Disable-PSSessionConfiguration -Name *`. In Windows PowerShell 3.0 and later releases, `Disable-PSRemoting` is the equivalent of `Set-PSSessionConfiguration -Name <Configuration name> -AccessMode Local`

Example 1: Prevent remote access to all session configurations

```
Disable-PSRemoting
```

```
WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these
 steps:
    1. Stop and disable the WinRM service.
    2. Delete the listener that accepts requests on any IP address.
```

```
   3. Disable the firewall exceptions for WS-Management communications.
   4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
      members of the Administrators group on the computer.
```

Example 2: Prevent remote access to all session configurations without confirmation prompt

```
Disable-PSRemoting -Force
```

```
WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these
 steps:
   1. Stop and disable the WinRM service.
   2. Delete the listener that accepts requests on any IP address.
   3. Disable the firewall exceptions for WS-Management communications.
   4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
      members of the Administrators group on the computer.
```

---------- Example 3: Effects of running this cmdlet ----------

```
Disable-PSRemoting -Force
New-PSSession -ComputerName localhost
```

```
WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these steps:
   1. Stop and disable the WinRM service.
   2. Delete the listener that accepts requests on any IP address.
   3. Disable the firewall exceptions for WS-Management communications.
   4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
      members of the Administrators group on the computer.

New-PSSession : [localhost] Connecting to remote server localhost failed with the following error
```

```
 message : Access is denied. For more information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:1
+ New-PSSession -ComputerName localhost -ConfigurationName PowerShell.6
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
+ CategoryInfo          : OpenError: (System.Management.A\u2026tion.RemoteRunspace:RemoteRunspace)
 [New-PSSession], PSRemotingTransportException
+ FullyQualifiedErrorId : AccessDenied,PSSessionOpenFailed
```

Example 4: Effects of running this cmdlet and Enable-PSRemoting

```
Disable-PSRemoting -Force
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -AutoSize

Enable-PSRemoting -Force
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -AutoSize

Name                       Permission
----                       ----------
microsoft.powershell          NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell.workflow NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell32        NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.ServerManager       NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
WithProfile                   NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed

Name                       Permission
----                       ----------
microsoft.powershell          BUILTIN\Administrators AccessAllowed
microsoft.powershell.workflow BUILTIN\Administrators AccessAllowed
microsoft.powershell32        BUILTIN\Administrators AccessAllowed
microsoft.ServerManager       BUILTIN\Administrators AccessAllowed
WithProfile                   BUILTIN\Administrators AccessAllowed
```

The `Enable-PSRemoting` cmdlet re-enables remote access to all PowerShell session endpoint configurations on the computer. The Force parameter suppresses all user prompts and restarts the WinRM service without prompting. The new output shows that the AccessDenied security descriptors have been removed from all session configurations.

Example 5: Loopback connections with disabled session endpoint configurations

```
Disable-PSRemoting -Force
New-PSSession -ComputerName localhost

WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these steps:
    1. Stop and disable the WinRM service.
    2. Delete the listener that accepts requests on any IP address.
    3. Disable the firewall exceptions for WS-Management communications.
    4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
       members of the Administrators group on the computer.

New-PSSession : [localhost] Connecting to remote server localhost failed with the following error message : Access is
denied. For more information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:1
+ New-PSSession -ComputerName localhost
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : OpenError: (System.Manageme....RemoteRunspace:RemoteRunspace) [New-PSSession], PSRemotin
   gTransportException
    + FullyQualifiedErrorId : AccessDenied,PSSessionOpenFailed

New-PSSession -ComputerName localhost -EnableNetworkAccess

Id Name       Transport ComputerName  ComputerType   State   ConfigurationName   Availability
 -- ----       --------- ------------  ------------   -----   -----------------   ------------
 1  Runspace1  WSMan     localhost     RemoteMachine  Opened  powershell.6          Available
```

The first use of `New-PSSession` attempts to create a remote session to the local machine. This type of connection goes through the network stack and is not a

loopback. Consequently, the connection attempt to the disabled endpoint fails with an Access is denied error.

The second use of `New-PSSession` also attempts to create a remote session to the local machine. In this case, it succeeds because it is a loopback connection that

bypasses the network stack.

A loopback connection is created when the following conditions are met:

- The computer name to connect to is 'localhost'.

- No credentials are passed in. Current logged in user (implicit credentials) is used for the

connection. - The EnableNetworkAccess switch parameter is used.

For more information on loopback connections, see New-PSSession (New-PSSession.md)document.

Example 6: Prevent remote access to session configurations that have custom security descriptors

```
Register-PSSessionConfiguration -Name Test -FilePath .\TestEndpoint.pssc -ShowSecurityDescriptorUI -Force
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -Wrap

Disable-PSRemoting -Force
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -Wrap
New-PSSession -ComputerName localhost -ConfigurationName Test

Name                    Permission
----                   ----------
microsoft.powershell       BUILTIN\Administrators AccessAllowed
Test                   NT AUTHORITY\INTERACTIVE AccessAllowed, BUILTIN\Administrators AccessAllowed,
DOMAIN01\User01 AccessAllowed
```

```
WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these steps:
    1. Stop and disable the WinRM service.
    2. Delete the listener that accepts requests on any IP address.
    3. Disable the firewall exceptions for WS-Management communications.
    4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
       members of the Administrators group on the computer.

Name                       Permission
----                       ----------
microsoft.powershell         NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
Test                       NT AUTHORITY\NETWORK AccessDenied, NTAUTHORITY\INTERACTIVE AccessAllowed,
BUILTIN\Administrators AccessAllowed, DOMAIN01\User01 AccessAllowed



 [Server01] Connecting to remote server failed with the following error message : Access is denied. For more information,
see the about_Rem
ote_Troubleshooting Help topic.
      + CategoryInfo                    : OpenError: (System.Manageme....RemoteRunspace:RemoteRunspace) [],
PSRemotingTransportException
+ FullyQualifiedErrorId : PSSessionOpenFailed
```

Now the `Get-PSSessionConfiguration` and `Format-Table` cmdlets shows that an AccessDenied security descriptor for all network users is added to all session configurations, including the Test session configuration. Although the other security descriptors are not changed, the "network_deny_all" security descriptor takes precedence. This is illustrated by the attempt to use `New-PSSession` to connect to the Test session configuration.

Example 7: Re-enable remote access to selected session configurations

```
Disable-PSRemoting -Force
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -AutoSize

Set-PSSessionConfiguration -Name Microsoft.ServerManager -AccessMode Remote -Force
```

```
Get-PSSessionConfiguration | Format-Table -Property Name, Permission -AutoSize

WARNING: Disabling the session configurations does not undo all the changes made by the Enable-PSRemoting
 or Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes by following these steps:
    1. Stop and disable the WinRM service.
    2. Delete the listener that accepts requests on any IP address.
    3. Disable the firewall exceptions for WS-Management communications.
    4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to
       members of the Administrators group on the computer.

Name                          Permission
----                          ----------
microsoft.powershell          NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell.workflow NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell32        NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.ServerManager       NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
WithProfile                   NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed

Name                          Permission
----                          ----------
microsoft.powershell          NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell.workflow NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.powershell32        NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
microsoft.ServerManager       BUILTIN\Administrators AccessAllowed
WithProfile                   NT AUTHORITY\NETWORK AccessDenied, BUILTIN\Administrators AccessAllowed
```

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/disable-psremoting?view=powershell-5.1&WT.mc_id=ps-gethelp

Disable-PSSessionConfiguration

Enable-PSRemoting

Get-PSSessionConfiguration

New-PSSession

Register-PSSessionConfiguration

Set-PSSessionConfiguration

Unregister-PSSessionConfiguration

WSMan Provider