Validate Windows Server with Test-WindowsBestPractices Cmdlet

Overview

When deploying a new Pure Storage FlashArray it is important to make sure that Microsoft Windows Server is configured properly and set to the recommended best practice settings for Multipath-IO, TRIM/UNMAP support and checking that Offloaded Data Transfer (ODX) is enabled. In order to make this easy a Windows PowerShell cmdlet, Test-WindowsBestPractices, was introduced in the Pure Storage PowerShell Toolkit (Toolkit).

Windows Server Version Support

The Test-WindowsBestPractices PowerShell cmdlet only works with Microsoft Windows Server 2012, 2012 R2 and 2016. It can also be used on Windows Server Core. The reason that the cmdlet is only supported under the latest versions of Windows Server is because of the various core PowerShell cmdlets that are used and only supported on the aforementioned versions of Windows Server.

How to Install to Toolkit

Open up an elevated Windows PowerShell session and enter the below commands.

```
An Untrusted Repository warning will display if the PowerShell Gallery (PSGallery) has not been added as a trusted repository. To check this information use the below cmdlets.

PS C:\> Set-PSRepository -Name PSGallery -InstallationPolicy Trusted
PS C:\> Get-PSRepository

Name InstallationPolicy SourceLocation
---- --------------- ---------------
PSGallery Trusted https://www.powershellgallery.com/api/v2/

PS C:\> Install-Module -Name PureStoragePowerShellToolkit
```

©2020 Copyright Pure Storage. All rights reserved.
Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its
InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from
'PSGallery'?  
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
PS C:\> Get-Module -ListAvailable "*Toolkit"

Directory: C:\Program Files\WindowsPowerShell\Modules

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     3.4.619.0  PureStoragePowerShellToolkit        {New-FlashArrayCapacityReport, Test-WindowsBestPractices, ...

PS C:\> Get-Command Test-WindowsBestPractices

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Function        Test-WindowsBestPractices                          3.4.619.0  PureStoragePowerShellToolkit

How to Use Test-WindowsBestPractices

Using the Test-WindowsBestPractices cmdlet is very easy. Open a Windows PowerShell session and type the name of the cmdlet and hit enter.

==========================================================================================================
Pure Storage Windows Server Best Practice Analyzer
==========================================================================================================

Host Information
==========================================================================================================
ChassisSerialNumber  : FCH1834J5CP
CollectedDateTime    : 10/26/2017 8:14:14 AM
Model                : UCSB-B200-M3
Name                 : SERVER01
NumberOfCores        : 20
NumberOfLogicalProcessors : 40
NumberOfProcessors   : 2
OSName               : Microsoft Windows Server 2016 Datacenter
OSSku                : 8
OSSuite              : 400
OSSuiteMask          : 400
OSVersion            : 10.0.14393
ProcessorFamily      : 179
ProcessorManufacturer : GenuineIntel
ProcessorName        : Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
SystemManufacturer   : Cisco Systems Inc

==========================================================================================================
Multipath-IO Verification

©2020 Copyright Pure Storage. All rights reserved.
PASS: Multipath-IO is installed.

MPIO Setting Verification

Current MPIO Configuration

PathVerificationState : Disabled
PathVerificationPeriod : 30
PDORemovePeriod : 30
RetryCount : 3
RetryInterval : 1
UseCustomPathRecoveryTime : Enabled
CustomPathRecoveryTime : 20
DiskTimeoutValue : 60

PASS: Microsoft Device Specific Module (MSDSM) is configured for Pure Storage FlashArray.
PASS: MPIO PDORemovePeriod passes Windows Server Best Practice check.
PASS: MPIO UseCustomPathRecoveryTime passes Windows Server Best Practice check.
PASS: MPIO CustomPathRecoveryTime passes Windows Server Best Practice check.
PASS: MPIO DiskTimeoutValue passes Windows Server Best Practice check.

TRIM/UNMAP Verification

PASS: Delete Notification Enabled

How to Read the Output

There are several sections of output that the cmdlet generates.

- Host Information -- Provides details on the current host being verified for best practices.
- Multipath-IO Verification -- Checks that Multipath-IO has been installed on the host.
- MPIO Setting Verification -- Checks and displays the current settings on the host for MPIO and then verifies those settings meet Pure Storage best practices.
- TRIM/UNMAP Verification -- Checks that Delete Notification is enabled.

Test-WindowsBestPractices Source

```powershell
# UNDER DEVELOPMENT -- Test-WindowsBestPractices
Test-WindowsBestPractices() {
    Clear-Host
    Write-Host '============================================================'
    Write-Host 'Pure Storage Windows Server Best Practice Analyzer'
    Import-Module ./PureStoragePowerShellToolkit.psd1
    Write-Host " Version: ${($_ = Get-Module -Name PureStoragePowerShellToolkit | Select-Object Version)}"
```
Write-Host '=================================================================='
Write-Host 'Host Information'
Get-SilComputer

function Write-Host ''
Write-Host '=================================================================='
Write-Host 'Multipath-IO Verification'
if (!(Get-WindowsFeature -Name 'Multipath-IO').InstalledStatus -eq 'Installed') {
    Write-Host 'PASS: Multipath-IO is installed.'
} else {
    $resp = Write-Warning 'FAIL: Multipath-IO Windows feature not installed. Would you like to install this feature?'
    if ($resp.ToUpper() -eq 'Y') {
        Add-WindowsFeature -Name Multipath-IO
    } else {
        Write-Warning 'Multipath-IO Windows feature has not been installed. Please add this feature manually.'
        break
    }
}

Write-Host ''
Write-Host '=================================================================='
Write-Host 'MPIO Setting Verification'
Write-Host '=================================================================='
Write-Host 'Current MPIO Configuration'
Get-MPIOSetting
Get-MPIOAvailableHW
Write-Host '=================================================================='

$BPFail = 0
$DSMs = Get-MPIOAvailableHW
ForEach ($DSM in $DSMs) {
    if (((($DSM).VendorId.Trim()) -eq 'PURE' -and (($DSM).ProductId.Trim()) -eq 'FlashArray')
        Write-Host 'PASS: Microsoft Device Specific Module (MSDSM) is configured for Pure Storage FlashArray.'
    $MPIO = Get-MPIOSetting
    if (($MPIO[4] -replace " ", "") -ceq 'PDORemovePeriod:30') {
        #30
        Write-Host 'PASS: MPIO PDORemovePeriod passes Windows Server Best Practice check.'
    } else {
        Write-Warning 'FAIL: MPIO PDORemovePeriod does NOT pass Windows Server Best Practice check.'
        $BPFail = $BPFail + 1
    }
        #Enabled
        Write-Host 'PASS: MPIO UseCustomPathRecoveryTime passes Windows Server Best Practice check.'
    } else {
        Write-Warning 'FAIL: MPIO UseCustomPathRecoveryTime does NOT pass Windows Server Best Practice check.'
        $BPFail = $BPFail + 1
    }
}
if (($MPIO[8] -replace "", "") -ceq 'CustomPathRecoveryTime:20') {
  #20
  Write-Host 'PASS: MPIO CustomPathRecoveryTime passes Windows Server Best Practice check.'
} else {
  Write-Warning 'FAIL: MPIO CustomPathRecoveryTime does NOT pass Windows Server Best Practice check.'
  $BPFail = $BPFail + 1
}
if (($MPIO[9] -replace "", "") -ceq 'DiskTimeoutValue:60') {
  #60
  Write-Host 'PASS: MPIO DiskTimeoutValue passes Windows Server Best Practice check.'
} else {
  Write-Warning 'FAIL: MPIO PDiskTimeoutValue does NOT pass Windows Server Best Practice check.'
  $BPFail = $BPFail + 1
}
}
if ($BPFail -gt 0) {
  $resp = Read-Host "Would you like to correct the failed MPIO settings?"
  if ($resp.ToUpper() -eq 'Y') {
    Set-MPIOSetting -NewPDORemovePeriod 30 -NewDiskTimeout 60 -CustomPathRecovery Enabled -NewPathRecoveryInterval 20
    Write-Host 'PASS: MPIO Configuration has been updated.'
  }
  else {
    Write-Warning 'No changes have been to MPIO settings. Please manually modify the settings to conform with Pure Storage Windows Server Best Practices.'
  }
}

Write-Host "
Write-Host 'TRIM/UNMAP Verification'
Write-Host '==================================================================='
if (!(Get-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\FileSystem' -Name 'DisableDeleteNotification') -eq 0) {
  Write-Host 'PASS: Delete Notification Enabled'
} else {
  Write-Warning 'Delete Notification Disabled. Pure Storage Best Practice is to enable delete notifications.'
}
}
Feedback

If you have feedback on the Test-WindowsBest Practices cmdlet please visit [https://github.com/purestorage-openconnect/powershell-toolkit](https://github.com/purestorage-openconnect/powershell-toolkit) and create a new Issue. The issues are reviewed and will be considered for future releases of the cmdlet and toolkit.