Windows PowerShell Get-Help on Cmdlet 'Test-AppLockerPolicy'

PS:\>Get-HELP Test-AppLockerPolicy -Full

NAME

Test-AppLockerPolicy

SYNOPSIS

Specifies the AppLocker policy to determine whether the input files will be allowed to run for a given user.

SYNTAX

Test-AppLockerPolicy [-XmlPolicy] <String> [-Filter {Allowed | AllowedByDefault | Denied | DeniedByDefault}] -Packages <System.Collections.Generic.List`1[Microsoft.Windows.Appx.PackageManager.Commands.AppxPackage]> [-User <String>] [<CommonParameters>]

Test-AppLockerPolicy [-XmlPolicy] <String> [-Filter {Allowed | AllowedByDefault | Denied | DeniedByDefault}] -Path <System.Collections.Generic.List`1[System.String]>
[-User <String>] [<CommonParameters>]

Test-AppLockerPolicy [-PolicyObject] <AppLockerPolicy> [-Filter {Allowed | AllowedByDefault | Denied | DeniedByDefault}] -Path
<System.Collections.Generic.List`1[System.String]> [-User <String>] [<CommonParameters>]

DESCRIPTION

The Test-AppLockerPolicy cmdlet specifies the AppLocker policy to determine whether a list of files is allowed to run on the local computer for a specified user.

To test AppLocker rules for a nested group, a representative member of the nested group should be specified for the User parameter. For example, a rule that allows

the Everyone group to run calc.exe may not appear to apply correctly when the nested Finance group for the User parameter is specified. Instead, a representative

member of the Finance group should be specified for the User parameter.

PARAMETERS

-Filter <System.Collections.Generic.List`1[Microsoft.Security.ApplicationId.PolicyManagement.PolicyDecision]>

Specifies the policy decision by which to filter the output for each input file. The acceptable values for this parameter are: Allowed, Denied, DeniedByDefault,

or AllowedByDefault.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Packages <System.Collections.Generic.List`1[Microsoft.Windows.Appx.PackageManager.Commands.AppxPackage]>

Specifies a list of installed packaged applications, from which the file information is retrieved.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Path <System.Collections.Generic.List`1[System.String]>

Specifies the list of the file paths to test. Regular expressions are supported.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-PolicyObject <AppLockerPolicy>

Specifies the Applocker policy. Can be obtained from the Get-AppLockerPolicy or the New-AppLockerPolicy cmdlet.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-User <String>

Defines the user or group to be used for testing the rules in a specified AppLocker policy. The acceptable values for this parameter are:

- DNS user name (`domain\username`)

- User Principal Name (`username@domain.com`)

- SAM user name (`username`)

- Security identifier (`S-1-5-21-3165297888-301567370-576410423-1103`)

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-XmlPolicy <String>

Specifies the file path and name of the XML-formatted file that contains the AppLocker policy.

Required? true

Position? 0

Default value None

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

Microsoft.Security.ApplicationId.PolicyManagement.PolicyModel.AppLockerPolicy

AppLockerPolicy

OUTPUTS

Microsoft.Security.ApplicationId.PolicyManagement.AppLockerPolicyDecision

NOTES

------- Example 1: Report if programs are allowed to run -------

PS C:\> Test-AppLockerPolicy -XMLPolicy C:\Policy.xml -Path c:\windows\system32\calc.exe, C:\windows\system32\notepad.exe -User Everyone

This example reports if calc.exe and notepad.exe will be allowed to run for Everyone under the policy specified by C:\Policy.xml.

------ Example 2: List executables specified by no policy ------

PS C:\> Get-ChildItem C:\windows\system32*.exe | Test-AppLockerPolicy c:\Policy.xml -Filter DeniedByDefault

This example lists the executables under C:\Windows\System32 that everyone will be denied by the policy specified by C:\Policy.xml because there is no explicit rule

for the file.

Example 3: List executables specified by no policy to a text file

PS C:\> Get-AppLockerPolicy -Local | Test-AppLockerPolicy -Path C:\Windows\System32*.exe -User contoso\saradavis -Filter Denied | Format-List -Property | Set-Content

(?C:\temp\DeniedFiles.txt?)

This example gets the local AppLocker policy, uses the policy to determine which executables in C:\Windows\System32 that contoso\saradavis is explicitly denied access

to run, and then redirects the list to a text file.

----- Example 4: Lists packages and test against a policy -----

PS C:\> Get-AppxPackage -AllUsers | Test-AppLockerPolicy -XmlPolicy .\SamplePolicy.xml

This example lists all the packages installed on this computer, for all the users, and tests them against a saved policy.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/applocker/test-applockerpolicy?view=windowsserver2022-ps&wt.mc_id=ps-gethelp

Get-AppLockerFileInformation

Get-AppLockerPolicy

New-AppLockerPolicy

Set-AppLockerPolicy