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

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

NAME

New-AppLockerPolicy

SYNOPSIS

Creates a new AppLocker policy from a list of file information and other rule creation options.

SYNTAX

New-AppLockerPolicy [-FileInformation] <System.Collections.Generic.List`1[Microsoft.Security.ApplicationId.PolicyManagement.PolicyModel.FileInformation]> [-AllowWindows] [-IgnoreMissingFileInformation] [-Optimize] [-RuleNamePrefix <String>] [-RuleType {Publisher | Path | Hash}] [-ServiceEnforcement <ServiceEnforcementMode>] [-User <String>] [-Xml] [<CommonParameters>]

DESCRIPTION

The New-AppLockerPolicy cmdlet uses a list of file information to automatically generate a list of rules for a given user or group. Rules can be generated based on publisher, hash, or path information.

Run the Get-AppLockerFileInformation cmdlet to create the list of file information.

By default, the output is an AppLockerPolicy object. If the Xml parameter is specified, the output will be the AppLocker policy as an XML-formatted string.

PARAMETERS

-AllowWindows [<SwitchParameter>]

Indicates that the AppLocker policy allows all local Windows components.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

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

Specifies a file that can contain publisher, path, and hash information. Some information may be missing, such as publisher information for an unsigned file.

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-IgnoreMissingFileInformation [<SwitchParameter>]

Specifies that, if a rule cannot be created for a file because of missing file information, then evaluation of the remaining file information will continue and a

warning log of the files skipped will be generated.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Optimize [<SwitchParameter>]

Specifies that similar rules will be grouped together.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-RuleNamePrefix <String>

Specifies a name to add as the prefix for each rule that is created.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

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

Specifies the type of rules to create from the file information. Publisher, path, or hash rules can be created from the file information. Multiple rule types may

be specified. Therefore, that there are backup rule types if the necessary file information is not available.

For example, if `Publisher, Hash` is specified for this parameter, then the hash rules are applied when publisher information is not available.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ServiceEnforcement <ServiceEnforcementMode>

Specifies whether the AppLocker policy for EXE and DLL rule collections applies to non-interactive processes. The acceptable values for this parameter are:

- NotConfigured

- Enabled

- ServicesOnly

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-User <String>

Specifies the user or group to which the rules are applied. 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

-Xml [<SwitchParameter>]

Specifies that the output of the AppLocker policy be as an XML-formatted string.

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

Microsoft.Security.ApplicationId.PolicyManagement.PolicyModel.FileInformation

OUTPUTS

Microsoft.Security.ApplicationId.PolicyManagement.PolicyModel.AppLockerPolicy

System.String

NOTES

---- Example 1: Create an AppLocker policy with allow rules ----

C:\PS>Get-ChildItem C:\Windows\System32*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -RuleType Publisher, Hash -User Everyone -RuleNamePrefix System32

Version	RuleCollections	RuleCollectionTypes
1	{Microsoft.Security.ApplicationId.Po...	{Exe}

This example creates an AppLocker policy that contains allow rules for all of the executable files in C:\Windows\System32. The policy contains publisher rules for

those files with publisher information and hash rules for those that do not. The rules are prefixed with `System32:` and the rules apply to the Everyone group.

---------- Example 2: Create an AppLocker policy ----------

C:\PS>Get-ChildItem C:\Windows\System32*.exe | Get-AppLockerFileInformation | New-AppLockerPolicy -AllowWindows -RuleType Path -User Everyone -Optimize -XML

<AppLockerPolicy Version="1"><RuleCollection Type="Exe" EnforcementMode="NotConfigured"><FilePathRule Id="31B2F340-016D

-11D2-945F-00C04FB984F9" Name="%SYSTEM32%*" Description="" 10 UserOrGroupSid="S-1-5-21-3165297888-301567370-576410423-

13" Action="cAllow"><Conditions><FilePathCondition Path="%SYSTEM32%*" /></Conditions></FilePathRule></RuleCollection>

</AppLockerPolicy>

This example creates an XML-formatted AppLocker policy for all of the executable files in `C:\Windows\System32`. The policy contains only path rules. The rules are

applied to the Everyone group. The Optimize parameter indicates that similar rules are grouped together where possible. The AppLocker policy trusts all local Windows

components.

-- Example 3: Create an AppLocker policy from audited events --

C:\PS>Get-AppLockerFileInformation -EventLog -LogPath "Microsoft-Windows-AppLocker/EXE and DLL" -EventType Audited | New-AppLockerPolicy -RuleType Publisher,Hash

-User domain\FinanceGroup -IgnoreMissingFileInformation | Set-AppLockerPolicy -LDAP

"LDAP://DC13.TailspinToys.com/CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=WingTipToys,DC=com"

This example creates a new AppLocker policy from the audited events in the local Microsoft-Windows-AppLocker/EXE and DLL event log. All of the rules will be applied

to the domain\FinanceGroup group. Publisher rules are created when the publisher information is available, and hash rules are created if the publisher information is

not available. If only path information is available for a file, then the file is skipped because the IgnoreMissingFileInformation parameter is specified, and the

file is included in the warning log. If the IgnoreMissingFileInformation parameter is not specified when file information is missing, then the cmdlet exits because it

cannot create the specified rule type. After the new AppLocker policy is created, the AppLocker policy of the specified Group Policy Object (GPO) is set. The existing

AppLocker policy in the specified GPO will be overwritten.

RELATED LINKS

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

Get-AppLockerFileInformation

Get-AppLockerPolicy

Set-AppLockerPolicy

Test-AppLockerPolicy