MICROSOFT WINDOWS SERVER 2016 STIG POWERSHELL DSC DOCUMENTATION

Version 1, Release 3

June 2020

Developed by DISA for the DoD

TABLE OF CONTENTS

Page

1. BACKGROUND

PowerShell DSC is a native facility within Microsoft Windows to define and enforce configurations. This package provides PowerShell DSC configurations that implement most of the Windows Server 2016 STIG. While the content has been tested during development, all possible system and environmental factors could not be tested. Before using this content in a production environment, please perform testing with the intended settings in your own test environment. There is no mandate to use this content; it is published as a resource to assist in the application of security guidance to your systems. Use it in the manner and to the extent that it assists with this goal.

Many Windows configuration settings can be applied using Group Policy. This DSC configuration does not leverage Group Policy. Therefore, this content is most useful for systems that are not using Group Policy or to manage a subset of configuration settings that are not being managed through Group Policy.

2. INSTALLATION

2.1 Modules

Unzip **`Win2016STIG-powershell.zip`**. In PowerShell, run the **`install.ps1`** script. This script downloads or copies the required modules into the **`C:\Program Files\WindowsPowerShell\Modules`** directory. See here for details.

Note: The install script temporarily trusts the Microsoft PowerShell Gallery to install dependent PowerShell modules from the Internet.

2.2 Certificates

The PowerShell DSC Certificate Import module requires certificates to be in an accessible location in the .CER format. If certificate management is desired, complete the following steps to export certificates in the .CER format:

1. Download InstallRoot from Cyber Exchange here.
2. Install it.
3. Run it.
4. Double-click on the certificate you want to export.
5. Choose the “Details” tab.
6. Verify the thumbprint matches your desired certificate.
7. Click “Copy to File”.
8. Click “Next”.
9. Choose a .CER format.
10. Click “Browse” to choose an export location and filename.
11. Click “Next”.
12. Click “Finish” to complete the export.
13. Continue the above process until all desired certificates are exported into .CER format.

3. CONFIGURATION

3.1 Simple

To apply the default STIG DSC configuration to the local machine only, run the **`enforce.ps1`** script in PowerShell to configure, compile, and enforce the STIG. To tailor the configuration, follow the steps in the next section.

3.2 Custom

In PowerShell, run **`New-STIGConfig -STIG Win2016STIG -Destination '.\config.psd1'`**. This creates a default PowerShell DSC configuration file that contains configuration data to define the machines to be configured, which configuration settings to manage, and the values for these settings. Edit this configuration file in a text editor to best suit each system's requirements as needed. For example, if you wanted to turn off STIG rule ID 87947, you would set it equal to **`$false`**. If you wanted to set STIG rule ID 87969's maximum password age to 90, you would set it to **`'90'`**.

```
stigrule_87947_Manage = $true
stigrule_87947_Telnet_Client_Ensure = 'Absent'
stigrule_87969_Manage = $true
stigrule_87969_AccountPolicies_Maximum_Password_Age_Maximum_Password_Age = '60'
```

For more information on configuration files, see here.

Note: For remote configurations, the dependent PowerShell modules must be installed on the remote systems before the configuration can be applied. The configuration does not currently install the required modules on remote systems.

3.3 Defaults

Some of the available settings are not managed by default. In cases where different rules apply depending on the domain role (domain controller, member server, or standalone), defaults are based on the standalone case. Other STIG rules have hardware requirements that cannot be managed by DSC, and while their configurations are provided here, they are not managed by default. Some settings require site-specific values, and these settings are not managed by default. Finally, certificate installation is not managed by default since the required certificates themselves must be obtained separately (see above).

4. COMPILATION

In PowerShell, run **`New-MOF -STIG Win2016STIG -ConfigurationData '.\config.psd1' -OutputPath '.\dsc'`**. This compiles a managed object format (MOF) file from your configuration data and desired STIG. MOF is the format that the local configuration manager (LCM) requires to enact configurations.

5. ENFORCEMENT

In PowerShell, run **`Start-DSC -Path '.\dsc'`**. This starts the enforcement of the DSC configuration by applying the configuration to the configured nodes. From this point on, the LCM will monitor according to the provided MOF file. The LCM can be configured to auto-correct as well as monitor. To do so, create and run the following configuration in PowerShell:

```
[DSCLocalConfigurationManager()]
configuration LCMConfig
{
    Node localhost
    {
        Settings
        {
            ConfigurationMode = 'ApplyAndAutoCorrect'
        }
    }
}
LCMConfig -OutputPath '.\dsc'
```

This compiles an MOF file of the LCM configuration. After running the above configuration, in PowerShell, run **`Set-DscLocalConfigurationManager -Path '.\dsc'`**. This applies the LCM MOF to the local LCM. For more LCM configuration settings, such as refresh frequency, see here.

6. DIAGNOSTICS

There are many diagnostic options available for PowerShell DSC with the included xDSCDiagnostics module. You can read about them here and here. Some of the most common and useful diagnostic commands include:

`Get-DscConfigurationStatus` for getting the current configuration status for a given node.

`Get-DscConfigurationStatus | Get-XDscConfigurationDetail` for getting verbose details of the status.

`Get-xDscOperation -Newest 20` for listing recent DSC operations to look for failures.

`Trace-xDscOperation -SequenceId 2` for listing events of the second most recent DSC operation.

`New-xDscDiagnosticsZip` for gathering diagnostics from the machine about the DSC state for support.

7. CONTENT EXTRACTION

This compliance extraction methodology returns results based on a system's compliance with the enforcement content. This may be different from STIG compliance. For example, multiple values may be allowed by the STIG but will be marked as "fail" if the value does not match the single exact value in the enforcement content. Additionally, if a value is customized in such a way as to violate a STIG rule, it will be marked as "pass" since it matches the enforcement content's expected value.

Once the local configuration manager is managing the state of your system after a successful configuration application, content extraction of the configuration results into XCCDF results can be performed. To generate XCCDF results, in PowerShell, run:

```
Write-XCCDF-Results -STIG
'\path\to\U_MS_Windows_Server_2016_STIG_V1R8_Manual-xccdf.xml' -Path
'\path\where\to\write\results.xml'
```

The above command writes the XCCDF results file at the **`-Path`** specified using the manual **`-STIG`** provided.

Note: The STIG provided above should match the STIG release and version number for which the PowerShell DSC content is built.

8. OTHER CONSIDERATIONS

8.1 Rebooting

Some DSC resources may request a reboot after applying. These requests stop the enforce script from running until a reboot has occurred and the local configuration manager resumes automatically. There are two solutions to handle these reboots: reboot and allow the LCM to continue in the background upon reboot or run the enforce script again without rebooting.

8.2 Consistency Checks in Progress

If the local configuration manager is in the process of performing a consistency check while attempting to run the enforce script, an error will occur mentioning a consistency check in progress. Wait until the check is complete to try again.