MICROSOFT WINDOWS SERVER 2022 STIG ANSIBLE DOCUMENTATION

Version 1, Release 1

07 February 2023

Developed by DISA for the DOD

TABLE OF CONTENTS

1. BACKGROUND

Ansible is an open source, cross-platform configuration management solution used to define and enforce system and application configurations. This package provides Ansible configurations that implement most of the Microsoft Windows Server 2022 STIG. While the content has been tested during development, all possible system and environmental factors could not be tested. Before applying this content in a production environment, users are advised to test with the intended settings in their 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 the user's 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 Ansible 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.

Ansible does not run natively on Windows. This documentation assumes the user will be running Ansible on a Red Hat Enterprise Linux 8 host machine and targeting the Microsoft Windows Server 2022 machine via WinRM.

This package uses basic authentication as a simple example of connecting to a Windows server via WinRM. For production use, consider configuring one of the more secure connection methods available. Refer here[1] for more information on connection methods.

[1] https://docs.ansible.com/ansible/latest/os_guide/windows_winrm.html#winrm-authentication-options

2. INSTALLATION

The following instructions are for standalone installation using ansible-playbook for testing. A production environment may also use Ansible Tower. Refer here[2] for details.

2.1 Installing Ansible

Newer versions of Ansible are in the Red Hat Enterprise Linux 8 Extra Packages for Enterprise Linux (EPEL) repository. To install it, run the following:

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
sudo yum install ansible
```

For other installation methods, refer here[3].

2.2 Installing Required Python Packages

Ansible uses the **pywinrm** package to communicate with Windows servers over WinRM. It is not installed by default with the Ansible package but can be installed by running the following:

```
pip3 install pywinrm
```

If pip is not installed, run the following:

```
sudo yum install python3-pip
python3 -m pip install --upgrade pip
```

2.3 Windows WinRM Setup

If needed, perform additional setup of WinRM on the target Windows Server 2022 machines by following the official Ansible documentation here[4].

2.4 Extracting

Unzip the **win2022STIG-ansible.zip**.

[2] https://www.ansible.com/products/controller
[3] https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installation-guide
[4] https://docs.ansible.com/ansible/latest/os_guide/windows_setup.html

3. CONFIGURATION

3.1 Simple

To apply the default STIG Ansible configuration to the local machine only, run the **`enforce.sh`** script to enforce the STIG.

The included hosts file is prepopulated with dummy values for **`ip_address, username, and password`**. Replace the values provided with the login details for the target Windows Server 2022 system.

To tailor the configuration, follow the steps in the next section.

3.2 Custom

To customize, create a YAML (.yml) file containing just the variables to customize from the variables named in the **`roles/win2022STIG/defaults/main.yml`** file. This file contains configuration data to define which configuration settings to manage and the values for these settings. Edit the newly created configuration file in a text editor to best suit each system's requirements as needed. For example, to turn off STIG rule ID 254273, set the "Manage" attribute to **`False`**. To set STIG rule ID 230369's minimum password length to 20, set the "**`_Maximum_Password_Age_Value`**" attribute to **`20`**.

```
win2022STIG_stigrule_254273_Manage: False
win2022STIG_stigrule_254273_Telnet_Client_State: absent

win2022STIG_stigrule_254289_Manage: True
win2022STIG_stigrule_254289_Maximum_Password_Age_Value: 20
```

To use the newly created, custom variables file, edit **`site.yml`** to include it. Refer to the highlighted (third and fourth) lines to add below:

```
- hosts: windows
  gather_facts: no
  vars_files:
    - /path/to/custom/vars.yml
  roles:
  - win2022STIG
```

For more information on variables, refer here[5]. For more information on YAML, refer here[6].

[5] https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html

[6] https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html

4. COMPLIANCE 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. If a value is customized in such a way to violate a STIG rule, it will be marked as “pass” because it matches the enforcement content’s expected value.

At the completion of a successful Ansible playbook play, the configuration results can be extracted into XCCDF results via an Ansible callback plugin. Use of this plugin can be controlled by modifying the following variable in the ansible.cfg file to include the name of the plugin:

```
[defaults]
callback_whitelist = stig_xml
```

Configuration of the plugin is controlled by creating/modifying the following environment variables:

- **`export STIG_PATH=/path/to/stig/ U_MS_Windows_Server_2022_STIG_V1R1_Manual-xccdf.xml`**
- **`export XML_PATH=/path/where/to/write/results.xml`**

The above environment variables control the plugin writing the XCCDF results to the file **`XML_PATH`** using the STIG at path **`STIG_PATH`**. The XCCDF results file is output by default to **`/tmp/tmpxxxxxx/xccdf-results.xml`**, where **`tmpxxxxxx`** is a randomly generated folder.

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

Ansible provides a means of checking compliance without enforcement called **`--check`** (i.e., “dry run”). To use this mode, run the following:

```
ansible-playbook -v -b -i /dev/null --check site.yml
```