SUSE LINUX ENTERPRISE SERVER (SLES) 15 STIG ANSIBLE DOCUMENTATION

Version 1, Release 10

27 April 2023

Developed by DISA for the DOD

TABLE OF CONTENTS

Page

1. BACKGROUND 1
2. INSTALLATION 2
2.1 Installing Ansible 2
2.2 Extracting 2
3. CONFIGURATION 3
3.1 Simple 3
3.2 Custom 3
4. COMPLIANCE EXTRACTION 4

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 SUSE Linux Enterprise Server (SLES) 15 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, perform testing with the intended settings in the 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 an individual's systems. Use it in the manner and to the extent that it assists with this goal.

2. INSTALLATION

The following instructions are for standalone installation using ansible-playbook for testing purposes. A production environment may additionally use the Ansible Automation Platform (formerly Ansible Tower). See here for details.

2.1 Installing Ansible

Ansible can be installed with pip, the Python package manager. For detailed instructions, see here.

To install it, run the following:

```
pip install ansible==2.10.7
```

For other installation methods, see here.

2.2 Extracting

Unzip the **sles15STIG-ansible.zip**.

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. Additionally, note that Ansible will refuse to reboot the local machine automatically. 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/sles15STIG/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 234825, set the "Manage" attribute equal to **`False`**. To set STIG rule ID 234891's max password lifetime to 30 days, set the "**`sles15STIG_stigrule_234891__etc_login_defs_Line`**" attribute to **`'PASS_MAX_DAYS 60'`**.

```
sles15STIG_stigrule_234825_Manage: False
sles15STIG_stigrule_234825__etc_login_defs_Line: 'ENCRYPT_METHOD SHA512'

sles15STIG_stigrule_234891_Manage: True
sles15STIG_stigrule_234891__etc_login_defs_Line: 'PASS_MAX_DAYS 30'
```

To use the newly created, custom variables file, edit **`site.yml`** to include it. See the highlighted lines to add below:

```
- hosts: localhost
  gather_facts: no
  vars_files:
    - /path/to/custom/vars.yml
  roles:
  - sles15STIG
```

For more information on variables, see here. For more information on YAML, see here.

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. Additionally, if a value is customized in such a way to violate a STIG rule it will be marked as "pass" since it matches the enforcement content's expected value.

At the completion of a successful Ansible playbook play content extraction of the configuration results into XCCDF results can be performed via an Ansible callback plugin. Use of this plugin can be controlled via modification of the follow variable in the ansible.cfg file to include the name of the plugin to use:

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

Configuration of the plugin is controlled via creation/modification of the following environment variables:

- **`export STIG_PATH=/path/to/stig/U_SLES_15_STIG_V1R10_Manual-xccdf.xml`**
- **`export XML_PATH=/path/where/to/write/results.xml`**

The above environmental 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 that the Ansible content is built for.

Ansible provides means of checking compliance without enforcement called **`--check`** (aka "dry run"). To use this mode, run the following:

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