CISCO IOS XE ROUTER RTR AND NDM STIG ANSIBLE DOCUMENTATION

Version 2, Release 1

October 2020

Developed by DISA for the DoD

TABLE OF CONTENTS

Page

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 Cisco IOS XE Router RTR and NDM STIGs. 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.

2. INSTALLATION

The following instructions are for stand-alone installation using ansible-playbook[1] for testing purposes. A production environment may additionally use Ansible Tower. See here[2] for details.

2.1 Online Installation

2.1.1 Installing Ansible

Newer versions of Ansible are in the Red Hat Enterprise Linux 7 EPEL[3] repository. To install it, run the following:

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo yum install ansible
```

For other installation methods, see here[4].

[1] https://docs.ansible.com/ansible/2.6/user_guide/playbooks_intro.html
[2] https://www.ansible.com/products/tower
[3] https://fedoraproject.org/wiki/EPEL
[4] https://docs.ansible.com/ansible/2.6/installation_guide/intro_installation.html#installation-guide

2.2 Offline Installation

2.2.1 Downloading Packages

On a similar system with internet access download the following packages automatically via the following command: **`sudo yum install --downloadonly --downloaddir=rpms ansible`**
If a similar system is not available, download the following packages manually:

```
Package                Arch    Version          Repository   Size
================================================================
Ansible                noarch  2.9.1-1.el7      epel         17 M
PyYAML                 x86_64  3.10-11.el7      rhel7-os    153 k
libyaml                x86_64  0.1.4-11.el7_0   rhel7-os     55 k
python-babel           noarch  0.9.6-8.el7      rhel7-os    1.4 M
python-cffi            x86_64  1.6.0-5.el7      rhel7-os    218 k
python-enum34          noarch  1.0.4-1.el7      rhel7-os     52 k
python-httplib2        noarch  0.9.2-0.2.el7    epel        115 k
python-idna            noarch  2.4-1.el7        rhel7-os     94 k
python-jinja2          noarch  2.7.2-4.el7      rhel7-os    519 k
python-markupsafe      x86_64  0.11-10.el7      rhel7-os     25 k
python-paramiko        noarch  2.1.1-9.el7      rhel7-os    269 k
python-ply             noarch  3.4-11.el7       rhel7-os    123 k
python-pycparser       noarch  2.14-1.el7       rhel7-os    105 k
python2-cryptography   x86_64  1.7.2-2.el7      rhel7-os    503 k
python2-jmespath       noarch  0.9.0-1.el7      epel         39 k
python2-pyasn1         noarch  0.1.9-7.el7      rhel7-os    100 k
sshpass                x86_64  1.06-1.el7       epel         21 k
```

This package list is based on a minimal installation of Red Hat Enterprise Linux 7. There may be slight variations between versions and distributions that would require additional packages. If any required packages are missing, the package manager will report this when attempting installation.

The Ansible package is available here[5]. EPEL packages are available here[6]. All other packages can be found here[7].

2.2.2 Installing Packages

The downloaded packages should be transferred to the offline system via authorized removable media or other approved methods. Packages can be installed together at once from a folder (here called rpms) via the following command:

```
sudo yum install rpms/*
```

Once complete, all packages should be installed. Verify ansible works by running the following:

[5] https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/
[6] https://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/
[7] http://mirror.centos.org/centos/7/os/x86_64/Packages/ -- Note that these are CentOS packages that are widely available and compatible with RHEL, but officially supported RHEL packages should be preferred for RHEL systems.

```
ansible --version
```

Ensure that the ansible version is 2.9.1 (or newer).

2.3 Extracting Content

Unzip the **iosxeSTIG-ansible.zip**.

3. CONFIGURATION

3.1 Simple

To apply the default STIG Ansible configuration, run the **enforce.sh** script to enforce the STIG.

The **enforce.sh** script requires a username argument, which is the remote username that Ansible will attempt to logon and run as after prompting for a password. It may take the resolvable name or IP address of the desired system to automatically generate a hosts file to use. Also, the **-i** flag accepts a previously prepared hosts file (see section 5.1) and the **-c** flag allows a check-only mode to be run (see section 4). For example:

```
$ sh enforce.sh jsmith 192.0.2.10
```

This generates a hosts file for **192.0.2.10** and attempts to run as **jsmith** after prompting for a password. The STIG Ansible playbook is then run against the system specified in the hosts file.

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/iosxeSTIG/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 215840, you would set the "Manage" variable to **False**. To set STIG rule ID 215807's session limit for all line vty sections to **3**, you would set the **iosxeSTIG_stigrule_105327_session_limit_for_all_line_vty_sections_Lines** variable to **'session-limit 3'**.

```
iosxeSTIG_stigrule_215840_Manage: False
iosxeSTIG_stigrule_215840_service_timestamps_log_datetime_localtime_Lines:
  - clock timezone EST -5 0

iosxeSTIG_stigrule_215807_Manage: True
iosxeSTIG_stigrule_215807_session_limit_for_all_line_vty_sections_Lines:
  - session-limit 3
```

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

```
- hosts: all
  gather_facts: no
  vars_files:
    - /path/to/custom/vars.yml
  roles:
  - iosxeSTIG
```

For more information on variables, see here[8]. For more information on YAML, see here[9].

[8] https://docs.ansible.com/ansible/2.7/user_guide/playbooks_variables.html
[9] 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. 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/stigs/stigs_are_here`**
- **`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 **`./xccdf-results.xml`**

Note: the STIG provided above should match the STIG release and version number that the Ansible content is built for. A copy of the STIG is provided in **`roles/iosxeSTIG/files`**.

Ansible provides means of checking compliance without enforcement called **`--check`** (aka "dry run"). To use this mode, use the enforce script with the **`-c`** flag. For example:

```
$ sh enforce.sh -c jsmith 192.0.2.10
```

5. OTHER CONSIDERATIONS

5.1 Configuring a Hosts File

To create a custom Ansible hosts entry, edit (or create) a **./hosts** file and put the resolvable names or IP addresses of the Cisco IOS-XE systems you want Ansible to manage there. For example:

```
192.0.2.10 ansible_connection=network_cli ansible_network_os=ios
```

For more information, see here[10].

5.2 Saving Configuration

Additional functionality has been added to the playbook to enable saving the running-configuration to the startup-configuration at the end of the play if and only if any of the tasks cause a configuration change. This functionality is disabled by default and can be enabled by setting the variable **iosxeSTIG_save_configuration_Manage** to **True**.

5.3 Solving Command Timeouts

For particularly slow Cisco IOS XE systems and/or configurations, you may encounter errors involving command timeouts. The default timeout for Ansible is 10 seconds. To increase this, either edit **/etc/ansible/ansible.cfg** or create a local **./ansible.cfg** with the following:

```
[persistent_connection]
command_timeout = 30
```

For more information, see here[11].

5.4 Error Handling

By default for enforcement, errors are blocking causing the run to fail on first error. By default for compliance extraction, errors are ignored and the run continues when an error is encountered. Error handling can be controlled via the **ignore_all_errors** variable in **iosxeSTIG.yml**. Set the variable to **True** to ignore all errors. The default is **False** to block on error.

[10] https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html
[11] https://docs.ansible.com/ansible/latest/network/user_guide/network_debug_troubleshooting.html#timeouts

5.5 Duplicates with Compliance Extraction

Due to effective duplicate STIG rules in the STIG, compliance extraction needs to be aware of which rules are duplicates of which other rules. This information is stored in **`roles/iosxeSTIG/files/duplicates.json`**. These values should not need to be changed unless it is desired to control how duplicates are handled during compliance extraction.

5.6 Availability of Configuration Options

Based on the particulars of the system under configuration, which may include factors such as the specific software version, license, or hardware platform, some configuration options enforced by this content may not be available. In this case, an error similar to the following will be encountered:

```
TASK [iosxeSTIG : stigrule_105391_password_min_len]
*************************************************
An exception occurred during task execution. To see the full
traceback, use -vvv. The error was: CiscoIOSXE(config)#
fatal: [x.x.x.x]: FAILED! => {"changed": false, "module_stderr":
"Traceback (most recent call last):\n
…
: aaa common-criteria policy PasswordPolicy\r\n
^\r\n% Invalid input detected at '^'
marker.\r\n\r\nCiscoIOSXE(config)#\n", "module_stdout": "", "msg":
"MODULE FAILURE\nSee stdout/stderr for the exact error", "rc": 1}
```

To work around unavailable configuration options, disable enforcement of these options by following the customization instructions in Section 3.2 and setting the “Manage” variable to **`False`** for the impacted rule(s).