

UNCLASSIFIED



CANONICAL UBUNTU 20.04 LTS STIG ANSIBLE DOCUMENTATION

Version 1, Release 9

26 July 2023

Developed by DISA for the DOD

UNCLASSIFIED

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 Canonical Ubuntu 20.04 LTS 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.

2. INSTALLATION

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

2.1 Installing Ansible

On Ubuntu, Ansible is available in the usual repo. To install it, run the following:

```
sudo apt install ansible
```

For further installation guidance, see [here](#)³.

2.2 Extracting

Unzip the `ubuntu2004STIG-ansible.zip`.

¹ https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html

² <https://www.ansible.com/products/tower>

³ https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu

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. Please note that you may need to set the `STIG_PATH` variable per section 4. To tailor the configuration, follow the steps in the next section.

3.2 Custom

To customize, create a YAML (.yaml) file containing just the variables to customize from the variables named in the `roles/ubuntu2004STIG/defaults/main.yaml` 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, if you wanted to turn off STIG rule ID 238202, you would set the "Manage" attribute equal to `False`. If you wanted to set STIG rule ID 238225's minimum password length to 20, you would set the `"_etc_security_pwquality_conf_Line"` attribute to `'minlen = 20'`.

```
Ubuntu2004STIG_stigrule_238202_Manage: False
Ubuntu2004STIG_stigrule_238202__etc_login_defs_Line: 'PASS_MAX_DAYS 60'

Ubuntu2004STIG_stigrule_238225_Manage: True
Ubuntu2004STIG_stigrule_238225__etc_security_pwquality_conf_Line: 'minlen = 20'
```

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

```
- hosts: localhost
  gather_facts: no
  vars_files:
    - /path/to/custom/vars.yaml
  roles:
    - ubuntu2004STIG
```

For more information on variables, see [here](https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html)⁴. For more information on YAML, see [here](https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html)⁵.

⁴ https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

⁵ 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 following 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_CAN_Ubuntu_20-04_LTS_STIG_V1R9_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/xccdf-results.xml`

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
```