AlgoSec Ansible Documentation

Release 0.0.1

Almog Cohen

Jan 23, 2019

Contents:

1 AlgoSec Ansible Role **1**
- 1.1 Requirements . . . 1
- 1.2 Installation . . . 1
- 1.3 Usage . . . 1
- 1.4 Documentation . . . 2
  - 1.4.1 How to build doc's locally? . . . 2
- 1.5 License . . . 2
- 1.6 Author Information . . . 2
- 1.7 Development . . . 2

2 Modules list **3**
- 2.1 algosec_define_application_flows . . . 3
  - 2.1.1 Synopsis . . . 3
  - 2.1.2 Requirements . . . 3
  - 2.1.3 Options . . . 4
  - 2.1.4 Return Values . . . 4
  - 2.1.5 Examples . . . 4
  - 2.1.6 Example For Application Flows JSON File . . . 4
- 2.2 algosec_add_single_application_flow . . . 5
  - 2.2.1 Synopsis . . . 5
  - 2.2.2 Requirements . . . 5
  - 2.2.3 Options . . . 5
  - 2.2.4 Examples . . . 5
- 2.3 algosec_provision_network_connectivity . . . 5
  - 2.3.1 Synopsis . . . 6
  - 2.3.2 Requirements . . . 6
  - 2.3.3 Options . . . 6
  - 2.3.4 Examples . . . 6
  - 2.3.5 Return Values . . . 7

3 Examples **9**
- 3.1 algosec_define_application_flows . . . 9
  - 3.1.1 Example For Application Flows JSON File . . . 10
- 3.2 algosec_add_single_application_flow . . . 10
- 3.3 algosec_provision_network_connectivity . . . 11

4 License **13**

5 Support 15

CHAPTER 1

AlgoSec Ansible Role

Ansible role to DevOps-ify network security management, leveraging AlgoSec's business-driven security policy management solution

Documentation available online at: http://algosec-ansible-role.readthedocs.io/en/latest/

1.1 Requirements

- This module is supported and fully tested under `python2.7` and `python3.6`.
- All modules of this role require environment:

```
pip install algosec --upgrade
pip install ansible marshmallow urllib3
```

1.2 Installation

The Ansible role can be installed directly from Ansible Galaxy by running:

```
ansible-galaxy install algosec.algosec
```

If the `ansible-galaxy` command-line tool is not available (usually shipped with Ansible), or you prefer to download the role package directly, navigate to the Ansible Galaxy role page and hit "Download".

Alternately, you can directly navigate to our GitHub repository.

1.3 Usage

Once installed, you can start using the modules included in this role in your ansible playbooks.

To quickly get up and running a simple example you can follow these steps:

1. Download and unzip locally the examples folder by clicking here.
2. Update authentication credentials in `vars/algosec-secrets.yml`.
3. Update your AlgoSec server IP in `inventory.ini`.
4. Update the arguments of the relevant modules in one of the playbooks (files with the `yml` extension).
5. Run `ansible-playbook -i inventory.ini <playbook-filename>.yml`.
6. You've made it!

1.4 Documentation

Documentation available online at: https://algosec-ansible-role.readthedocs.io/en/latest/

1.4.1 How to build doc's locally?

Using Docker, running from one folder outside of the project:

```
$ docker run -it -v $PWD/ansible-role-algosec/:/documents/ ivanbojer/spinx-with-rtd
$ cd docs
$ make html
```

Using Spinx:

```
$ cd docs
$ make html
```

Then see the `docs/_build` folder created for the html files.

1.5 License

MIT (see full license here)

1.6 Author Information

AlgoSec Official Website https://www.algosec.com/

1.7 Development

To kickoff local development, just use *pipenv*:

```
pipenv install
```

And to use the newly installed virtual environment just run:

```
pipenv shell
```

CHAPTER 2

Modules list

2.1 algosec_define_application_flows

New in version 0.3.0.

- *Synopsis*
- *Requirements*
- *Options*
- *Return Values*
- *Examples*
- *Example For Application Flows JSON File*

2.1.1 Synopsis

- Update application flows of an AlgoSec BusinessFlow application to match a requested configuration.
- Create, modify or delete application flows if needed.
- Apply the changes in BusinessFlow to automatically create a FireFlow change request.
- Optionally make sure that all defined flow pass the flow connectivity check on BusinessFlow

2.1.2 Requirements

- *algosec* can be obtained from PyPi https://pypi.python.org/pypi/algosec

2.1.3 Options

2.1.4 Return Values

2.1.5 Examples

```
---
- name: Update application flows of an AlgoSec BusinessFlow application
  hosts: algosec-server
  gather_facts: False

  roles:
    - role: algosec.algosec

  tasks:
  - name: Grab AlgoSec credentials from ansible-vault
    include_vars: 'algosec-secrets.yml'
    no_log: 'yes'

  - name: Set App flows on ABF using JSON configuration loaded from file
    # We use delegation to use the local python interpreter (and virtualenv if␣
↪enabled)
    delegate_to: localhost
    vars:
      flows_data: "{{ lookup('file','vars/application-flows.json')|from_json }}"

    algosec_define_application_flows:
      ip_address: "{{ ip_address }}"
      user: "{{ username }}"
      password: "{{ password }}"
      app_name: "{{ item.app_name}}"
      app_flows: "{{item.app_flows}}"
    with_items: "{{ flows_data.applications }}"
```

2.1.6 Example For Application Flows JSON File

```
{
  "applications": [
    {
      "app_name": "TEST",
      "app_flows": {
        "flow1": {
          "sources": ["HR Payroll server", "192.168.0.0/16"],
          "destinations": ["16.47.71.62"],
          "services": ["HTTPS"]
        },
        "flow2": {
          "sources": ["10.0.0.1"],
          "destinations": ["10.0.0.2"],
          "services": ["udp/501"]
        },
        "flow3": {
          "sources": ["1.2.3.4"],
          "destinations": ["3.4.5.6"],
          "services": ["SSH"]
```

(continues on next page)

(continued from previous page)

```
        }
      }
    },
    {
      "app_name": "ANOTHER-APP",
      "app_flows": {
        "new-flow": {
          "sources": ["1.2.3.4"],
          "destinations": ["3.4.5.6"],
          "services": ["SSH"]
        }
      }
    }
  ]
}
```

2.2 algosec_add_single_application_flow

New in version 0.1.0.

- *Synopsis*
- *Requirements*
- *Options*
- *Examples*

2.2.1 Synopsis

- Create a new application flow on AlgoSec BusinessFlow.
- Creation is skipped if the requested flow is a contained in one of the existing flows of the relevant application.

2.2.2 Requirements

- *algosec* can be obtained from PyPi https://pypi.python.org/pypi/algosec

2.2.3 Options

2.2.4 Examples

2.3 algosec_provision_network_connectivity

New in version 0.1.0.

- *Synopsis*
- *Requirements*
- *Options*
- *Examples*
- *Return Values*

2.3.1 Synopsis

- Provision network connectivity by creating a change request in AlgoSec FireFlow.
- No change request is created if traffic is already provisioned correctly.

2.3.2 Requirements

- *algosec* can be obtained from PyPi https://pypi.python.org/pypi/algosec

2.3.3 Options

2.3.4 Examples

```
---
- name: Create Traffic Change Request if needed
  hosts: algosec-server
  gather_facts: False

  roles:
    - role: algosec.algosec

  tasks:
  - name: Grab the credentials from ansible-vault
    include_vars: 'algosec-secrets.yml'
    no_log: 'yes'

  - name: Create Traffic Change Request
    # We use delegation to use the local python interpreter (and virtualenv if␣
↪enabled)
    delegate_to: localhost
    algosec_provision_network_connectivity:
      ip_address: "{{ ip_address }}"
      user: "{{ username }}"
      password: "{{ password }}"

      requestor: Almog Cohen
      email: almog@email.com
      traffic_lines:
        # This is an 'allow' traffic line
        - action: true
          sources: ["192.168.12.12", "123.123.132.123"]
          destinations: ["16.47.71.62", "234.234.234.234"]
          services: ["HTTPS", "http", "tcp/80", "tcp/51"]
```

(continues on next page)

(continued from previous page)

```
        # This is a drop traffic line
        - action: false
          sources: ["10.0.0.1"]
          destinations: ["10.0.1.0"]
          services: ["HTTPS"]

    register: result

  - name: Print the test results
    debug: var=result
```

2.3.5 Return Values

CHAPTER 3

Examples

3.1 algosec_define_application_flows

Match the application flows of an AlgoSec BusinessFlow application to a requested configuration

```
---
- name: Update application flows of an AlgoSec BusinessFlow application
  hosts: algosec-server
  gather_facts: False

  roles:
    - role: algosec.algosec

  tasks:
  - name: Grab AlgoSec credentials from ansible-vault
    include_vars: 'algosec-secrets.yml'
    no_log: 'yes'

  - name: Set App flows on ABF using JSON configuration loaded from file
    # We use delegation to use the local python interpreter (and virtualenv if␣
↪enabled)
    delegate_to: localhost
    vars:
      flows_data: "{{ lookup('file','vars/application-flows.json')|from_json }}"

    algosec_define_application_flows:
      ip_address: "{{ ip_address }}"
      user: "{{ username }}"
      password: "{{ password }}"
      app_name: "{{ item.app_name}}"
      app_flows: "{{item.app_flows}}"
    with_items: "{{ flows_data.applications }}"
```

3.1.1 Example For Application Flows JSON File

```
{
  "applications": [
    {
      "app_name": "TEST",
      "app_flows": {
        "flow1": {
          "sources": ["HR Payroll server", "192.168.0.0/16"],
          "destinations": ["16.47.71.62"],
          "services": ["HTTPS"]
        },
        "flow2": {
          "sources": ["10.0.0.1"],
          "destinations": ["10.0.0.2"],
          "services": ["udp/501"]
        },
        "flow3": {
          "sources": ["1.2.3.4"],
          "destinations": ["3.4.5.6"],
          "services": ["SSH"]
        }
      }
    },
    {
      "app_name": "ANOTHER-APP",
      "app_flows": {
        "new-flow": {
          "sources": ["1.2.3.4"],
          "destinations": ["3.4.5.6"],
          "services": ["SSH"]
        }
      }
    }
  ]
}
```

3.2 algosec_add_single_application_flow

Create new Application Flows on AlgoSec BusinessFlow

```
---
- name: Create a flow on an AlsogsecBusinessFlow App
  hosts: algosec-server
  gather_facts: False

  roles:
    - role: algosec.algosec

  tasks:
  - name: Grab the credentials from ansible-vault
    include_vars: 'algosec-secrets.yml'
    no_log: 'yes'

  - name: Create the flow on ABF
```

(continues on next page)

(continued from previous page)

```
    # We use delegation to use the local python interpreter (and virtualenv if␣
↪enabled)
    delegate_to: localhost
    algosec_add_single_application_flow:
      ip_address: "{{ ip_address }}"
      user: "{{ username }}"
      password: "{{ password }}"

      app_name: Payroll
      name: payroll-server-auth
      sources: ["192.168.12.12"]
      destinations: ["16.47.71.62", "16.47.71.63"]
      services: ["HTTPS", "tcp/23"]
```

3.3 algosec_provision_network_connectivity

Check and create traffic change requests with AlgoSec FireFlow.

```
---
- name: Create Traffic Change Request if needed
  hosts: algosec-server
  gather_facts: False

  roles:
    - role: algosec.algosec

  tasks:
  - name: Grab the credentials from ansible-vault
    include_vars: 'algosec-secrets.yml'
    no_log: 'yes'

  - name: Create Traffic Change Request
    # We use delegation to use the local python interpreter (and virtualenv if␣
↪enabled)
    delegate_to: localhost
    algosec_provision_network_connectivity:
      ip_address: "{{ ip_address }}"
      user: "{{ username }}"
      password: "{{ password }}"

      requestor: Almog Cohen
      email: almog@email.com
      traffic_lines:
        # This is an 'allow' traffic line
        - action: true
          sources: ["192.168.12.12", "123.123.132.123"]
          destinations: ["16.47.71.62", "234.234.234.234"]
          services: ["HTTPS", "http", "tcp/80", "tcp/51"]
        # This is a drop traffic line
        - action: false
          sources: ["10.0.0.1"]
          destinations: ["10.0.1.0"]
          services: ["HTTPS"]
```

(continues on next page)

(continued from previous page)

```
    register: result

  - name: Print the test results
    debug: var=result
```

CHAPTER 4

License

CHAPTER 5

Support

This template/solution is released under an as-is, best effort, support policy. These scripts should be seen as community supported and AlgoSec. will contribute our expertise as and when possible. We do not provide technical support or help in using or troubleshooting the components of the project through our normal support options such as AlgoSec support teams and backline support options. The underlying product used by the scripts or templates are still supported, but the support is only for the product functionality and not for help in deploying or using the template or script itself.

Unless explicitly tagged, all projects or work posted in our GitHub repository or sites other than our official Downloads page are provided under the best effort policy.

- search