Switchmap-NG Documentation

Release 0.0.1

Peter Harrison

Nov 02, 2017

Contents:

Introduction

`switchmap-ng` is Python 3 inventory system that reports and tabulates the status of network connected devices. The information reported includes:

1. Open Systems Interconnection model (OSI model) data such as:

2. Layer 1 information (Network port names, speed, state, neighbors)

3. Layer 2 information (VLANs, 802.1q trunk links)

4. Layer 3 information (ARP information)

5. System status

1.1 Features

`switchmap-ng` has the following features:

1. Open source.

2. Written in python, a modern language.

3. Easy configuration.

4. Threaded polling of devices for data. Fast.

5. Support for Cisco and Juniper gear. More expected to added with time.

6. Support for SNMPv2 and/or SNMPv3 for all configured network devices.

We are always looking for more contributors!

1.2 Inspiration

The project took inspiration from switchmap whose creator, Pete Siemsen, has been providing guidance.

1.3 Oversight

`switchmap-ng` is a student collaboration between:

1. The University of the West Indies Computing Society. (Kingston, Jamaica)

2. The University of Techology, IEEE Student Branch. (Kingston, Jamaica)

3. The Palisadoes Foundation http://www.palisadoes.org

And many others.

Installation

This section outlines how to install and do basic configuration of `switchmap-ng`.

2.1 Install Prerequisite Packages

`switchmap-ng` has the following requirements:

- python >= 3.5
- python3-pip

It will not work with lower versions.

2.1.1 Ubuntu / Debian / Mint

The commands for installing the dependencies are:

```
$ sudo apt-get -y install python3 python3-pip snmp
```

2.1.2 Centos / Fedora

The commands for installing the dependencies are:

```
$ sudo dnf -y install python3 python3-pip net-snmp-utils
```

2.2 Install Switchmap-NG

Installation is simple. Follow these steps

2.2.1 Clone the Repository

Now clone the repository and copy the sample configuration file to its final location.

```
$ git clone https://github.com/PalisadoesFoundation/switchmap-ng
$ cd switchmap-ng
```

2.2.2 Edit Configuration File

Edit the SNMP credential information in the configuration file.

```
$ cp examples/etc/config.yaml etc/config.yaml
$ vim etc/config.yaml

snmp_groups:
    - group_name: Corporate Campus
      snmp_version: 3
      snmp_secname: woohoo
      snmp_community:
      snmp_port: 161
      snmp_authprotocol: sha
      snmp_authpassword: testing123
      snmp_privprotocol: des
      snmp_privpassword: secret_password
```

2.2.3 Run Installation Script

Run the installation script. There are two alternatives:

Installing as a regular user

There are some things to keep in mind when installing *switchmap-ng* as a regular user.

1. Use this method if you don't have `root` access to your system.

2. The `switchmap-ng` daemons *will not* automatically restart on reboot using this method.

To make `switchmap-ng` run with your username, then execute this command:

```
$ maintenance/install.py
```

Installing as the "root" user

There are some things to keep in mind when installing *switchmap-ng* as the *root* user.

1. The `switchmap-ng` daemons *will* automatically restart on reboot using this installation method.

2. **Note**: Do not run setup using `sudo`. Use `sudo` to become the `root` user first.

To install `switchmap-ng` as the `root` user execute this command:

```
# maintenance/install.py
```

2.3 Testing Installation

There are a number of ways to test your installation.

2.3.1 Testing Polling

You can test your SNMP configuration and connectivity to your devices using the `switchmap-ng-cli` utility like this:

```
$ bin/switchmap-ng-cli test poller --all
```

2.3.2 Testing the Web Interface

You can test whether the API is working by visiting this url. (You will need to make adjustments if you installed the application on a remote server):

```
http://localhost:7000/switchmap-ng/
```

The Webserver help page provides the necessary steps to view switchmap on port 80 using Apache or Nginx

2.4 Next Steps

It is now time to review the various configuration options.

Configuration

The `examples/configuration` directory includes a sample file that can be edited. `switchmap-ng` assumes all files in this directory, or any other specified configuration directory, only contains `switchmap-ng` configuration files. Most user will only need to edit the three files supplied.

You must place your configuration file in the `etc/` directory as your permanent configuration file location.

3.1 Sample Configuration File

Here is a sample configuration file that will be explained later in detail. `switchmap-ng` will attempt to contact hosts with each of the parameter sets in the `snmp_group` section till successful.

```
main:
    log_directory: /home/switchmap-ng/log
    log_level: info
    cache_directory: /opt/switchmap-ng/cache
    agent_subprocesses: 20
    bind_port: 7000
    listen_address: 0.0.0.0
    hostnames:
      - 192.168.1.1
      - 192.168.1.2
      - 192.168.1.3
      - 192.168.1.4
    polling_interval: 86400

snmp_groups:
    - group_name: Corporate Campus
      snmp_version: 3
      snmp_secname: woohoo
      snmp_community:
      snmp_port: 161
      snmp_authprotocol: sha
      snmp_authpassword: testing123
```

```
        snmp_privprotocol: des
        snmp_privpassword: secret_password

    - group_name: Remote Sites
      snmp_version: 3
      snmp_secname: foobar
      snmp_community:
      snmp_port: 161
      snmp_authprotocol: sha
      snmp_authpassword: testing123
      snmp_privprotocol: aes
      snmp_privpassword: secret_password
```

3.1.1 The `main:` Section

This is the section of the configuration file that governs the general operation of `switchmap-ng`. Here is how it is configured.

Parameter	Description
`main:`	YAML key describing the server configuration.
`log_directory:`	The directory where `switchmap-ng` places its log files
`log_level:`	Defines the logging level. `debug` level is the most verbose, followed by `info`, `warning` and `critical`
`cache_directory:`	Location where data retrieved from devices will be stored.
`agent_subprocesses:`	The maximum number of subprocesses used to collect data from devices
`listen_address:`	IP address the API will be using. The default is `localhhost`. This should not need to be changed.
`bind_port:`	The TCP port the API will use. This should not need to be changed.
`hostnames:`	A list of hosts that will be polled for data.
`polling_interval:`	The frequency in seconds with which the poller will query devices

3.1.2 The `snmp_groups:` Section

This is the section of the configuration file that governs the SNMP credentials to be used to retrieve data from devices. You can have multiple groups, each with a separate `group_name`. This is how `switchmap-ng` uses this information.

1. `switchmap-ng` will attempt to use each set of group credentials until it is successful. It will skip devices that it cannot authenticate against or reach.

2. `switchmap-ng` will keep track of the most recently used credentials to successfully obtain data and will use these credentials first.

Parameter	Description
snmp_groups:	YAML key describing groups of SNMP authentication parameter. All parameter groups are listed under this key.
group_name:	Descriptive name for the group
snmp_version:	SNMP version. Must be present even if blank. Only SNMP versions 2 and 3 are supported by the project.
snmp_secname:	SNMP security name (SNMP version 3 only). Must be present even if blank.
snmp_community:	SNMP community (SNMP version 2 only). Must be present even if blank.
snmp_port:	SNMP Authprotocol (SNMP version 3 only). Must be present even if blank.
snmp_authprotocol:	SNMP AuthPassword (SNMP version 3 only). Must be present even if blank.
snmp_authpassword:	SNMP PrivProtocol (SNMP version 3 only). Must be present even if blank.
snmp_privprotocol:	SNMP PrivProtocol (SNMP version 3 only). Must be present even if blank.
snmp_privpassword:	SNMP PrivPassword (SNMP version 3 only). Must be present even if blank.
snmp_port:	SNMP UDP port

Command Line Interface (CLI)

This page outlines how to use the `switchmap-ng` command line interface (CLI)

4.1 Viewing `switchmap-ng` status

There are two important `switchmap-ng` daemons.

1. **poller:** Gets data from devices
2. **API:** Displays device data on web pages

You can get the status of each daemon using the following CLI commands:

4.1.1 Poller status

You can get the status of the poller using this command:

```
$ bin/switchmap-ng-cli show poller status
```

4.1.2 API status

You can get the status of the API using this command:

```
$ bin/switchmap-ng-cli show api status
```

4.2 Managing the `switchmap-ng` Daemons

You can manage the daemons using the CLI. Here's how:

4.2.1 Poller Management

The poller can be started, stopped and restarted using the following commands. Use the `--force` option only if the daemon may be hung.

```
$ bin/switchmap-ng-cli poller start

$ bin/switchmap-ng-cli poller stop
$ bin/switchmap-ng-cli poller stop --force

$ bin/switchmap-ng-cli poller restart
$ bin/switchmap-ng-cli poller restart --force
```

Note: You will need to do a restart whenever you modify a configuration parameter.

4.2.2 API Management

The API can be started, stopped and restarted using the following commands. Use the `--force` option only if the daemon may be hung.

```
$ bin/switchmap-ng-cli api start

$ bin/switchmap-ng-cli api stop
$ bin/switchmap-ng-cli api stop --force

$ bin/switchmap-ng-cli api restart
$ bin/switchmap-ng-cli api restart --force
```

Note: You will need to do a restart whenever you modify a configuration parameter.

4.3 Testing The Ability to Poll Devices

You will need to verify that the poller can access the hosts in the configuration.

4.3.1 Viewing Configured Hosts

You can view the configured hosts using this command.

```
$ bin/switchmap-ng-cli show hostnames
```

4.3.2 Testing Host Pollability

You can test a host using this command.

```
$ bin/switchmap-ng-cli test poller --hostname HOSTNAME
```

You can test all hosts using this command.

```
$ bin/switchmap-ng-cli test poller --all
```

4.4 Viewing `switchmap-ng` logs

When troubleshooting it is a good practice to view the `switchmap-ng` log files.

4.4.1 Poller logs

You can view the poller logs using this command:

```
$ bin/switchmap-ng-cli show poller logs
```

4.4.2 API logs

You can view the API logs using this command:

```
$ bin/switchmap-ng-cli show api logs
```

4.5 Viewing the `switchmap-ng` Configuration

You can view the configuration using this command:

```
$ bin/switchmap-ng-cli show configuration
```

Advanced Operation

The `switchmap-ng` CLI is meant for ease of use. This page shows some advanced features.

5.1 Operating the Poller as a System Daemon

This is the preferred mode of operation for **production systems**. This mode is automatically configured if you installed `switchmap-ng` using the `root` user.

Note: Sample `systemd` files can be found in the `examples/linux/systemd/` directory.

The `poller` can be started like this:

```
$ sudo systemctl start switchmap-ng-poller.service
```

The `poller` can be stopped like this:

```
$ sudo systemctl stop switchmap-ng-poller.service
```

You can get the status of the `poller` like this:

```
$ sudo systemctl status switchmap-ng-poller.service
```

You can get the `poller` to automatically restart on boot like this:

```
$ sudo systemctl enable switchmap-ng-poller.service
```

5.2 Operating the API as a System Daemon

This is the preferred mode of operation for production systems. This mode is automatically configured if you installed `switchmap-ng` using the `root` user.

Note: Sample `systemd` files can be found in the `examples/linux/systemd/` directory.

The `API` can be started like this:

```
$ sudo systemctl start switchmap-ng-api.service
```

The `API` can be stopped like this:

```
$ sudo systemctl stop switchmap-ng-api.service
```

You can get the status of the `API` like this:

```
$ sudo systemctl status switchmap-ng-api.service
```

You can get the `API` to automatically restart on boot like this:

```
$ sudo systemctl enable switchmap-ng-api.service
```

Webserver

You can access `switchmap-ng` on its default port 7000, however you may want to access it on port 80 by integrating it with an Apache or Nginx webserver. This page explains how.

6.1 Apache setup for `switchmap-ng`

`switchmap-ng` has sample configurations for the Apache webserver.

Run the following commands from the top directory of `switchmap-ng`

```
$ sudo cp examples/linux/apache/switchmap-ng-apache.conf /etc/apache2/conf-available
$ sudo ln -s /etc/apache2/conf-available/switchmap-ng-apache.conf /etc/apache2/conf-
↪enabled/switchmap-ng-apache.conf

# (Ubuntu only)
$ sudo a2enmod proxy_http
$ sudo systemctl restart apache2.service

# (RedHat / CentOS)
$ sudo systemctl restart httpd.service
```

You should now be able to access your `switchmap-ng` web pages using the following link.

```
http://SERVER_NAME/switchmap-ng/
```

6.2 Nginx setup for `switchmap-ng`

`switchmap-ng` has sample configurations for the Nginx webserver.

Run the following commands from the top directory of `switchmap-ng`

```
$ sudo cp examples/linux/nginx/switchmap-ng-nginx.conf /etc/nginx/sites-available
$ sudo ln -s /etc/nginx/sites-available/switchmap-ng-nginx.conf /etc/nginx/conf-
→enabled/switchmap-ng-nginx.conf
```

Note: Edit the `/etc/nginx/conf-enabled/switchmap-ng-nginx.conf` file and change the IP address of the server then restart Nginx.

```
$ systemctl restart nginx.service
```

You should now be able to access your `switchmap-ng` web pages using the following link.

```
http://SERVER_NAME/switchmap-ng/
```

Testing & Troubleshooting

Here's how you can test your installation of `switchmap-ng`.

7.1 Testing Operation After Installation

There are a number of steps to take to make sure you have installed `switchmap-ng` correctly. This section explains how to do basic testing before putting `switchmap-ng` into production.

7.1.1 Start the API Interactively

Start the `switchmap-ng` API interactively.

```
$ bin/switchmap-ng-api --start
```

7.1.2 Start the Poller Interactively

The poller will need to be running prior to testing.

```
$ bin/switchmap-ng-poller --start
```

7.1.3 Test Poller Functionality

Now that both the API and poller are running, it's time to test functionality by running the `bin/tools/test_installation.py` script

Here is an example of a successful test:

```
$ bin/tools/test_installation.py --all_hosts
Valid credentials found: hostname1
Valid credentials found: hostname2
$
```

You will see errors if none of the configured SNMP groups can be used to contact a host, or the host is not contactable. If this happens, check your configuration and the network access from your server to the remote host.

7.1.4 Test API Functionality

Testing the API is easy. Just visit the following URL:

```
http://hostname/switchmap-ng
```

A sample system startup script can be found in the `examples/linux/systemd/switchmap-ng-poller.service` file. Follow the instructions in the file to make changes to the startup operation of the `poller` daemon.

Note: There will be no visible output when the `poller` is running. The `poller` logs its status to the `log/switchmap-ng.log` file by default. You will be able to see this interaction dynamically by running the following command:

```
$ tail -f etc/switchmap-ng.log
```

7.2 Troubleshooting Using System Logs

There are different log files to check.

7.2.1 Troubleshooting the API

There will be no visible output when the `API` is running. The `API` logs its status to the `log/switchmap-ng-api.log` file by default. You will be able to see this interaction dynamically by running the following command:

```
$ tail -f etc/switchmap-ng-api.log
```

7.2.2 Troubleshooting the Poller

There will be no visible output when the `Poller` is running. The `Poller` logs its status to the `log/switchmap-ng.log` file by default. You will be able to see this interaction dynamically by running the following command:

```
$ tail -f etc/switchmap-ng.log
```

Community

Use these channels to communicate about the project.

8.1 Mailing list

The **user** mailing list is general discussion and support list for Gunicorn users.

- To **subscribe**, send an email to *TBD*
- To **unsubscribe**, send an email to *TBD*
- Finally, to post a message to the list use the address to *TBD*

The archive for this list can also be browsed online.

8.2 Issue Tracking

Bug reports, enhancement requests and tasks generally go in the Github issue tracker.

8.3 Security Issues

The security mailing list is a place to report security issues. Only developers are subscribed to it. To post a message to the list use the address to *TBD*.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search