**Pscan Version 0.4**

Developed by ALE SE Team

For support contact [kaveh.majidi@al-enterprise.com](mailto:kaveh.majidi@al-enterprise.com)

Pscan is a python tool to scan an ALE network for inactive ports and based on the defined criteria, Admin down the ports that have been inactive for specified number of days.

The tool uses OV (OmniVisa) API library for discovery and configuration changes. All the operations including data gathering and making configuration changes are done through OV and there is no direct access to switches. This method provides couple of advantages compared to direct interaction with switches.

1. The communication is happening only between a Linux/Windows machine (where the script is executing) and OV, resulting in a very small footprint to manage and secure.
2. Avoiding SSH connection to every single switch in the network will result in better performance and less SSH traffic in the network.
3. It’s easier to develop and change a tool which interact with a single OV database rather than reaching various types of switches.

There are some considerations when using OV API (OV database) as source of data for the tool. Pscan is dependent on OV and it requires OV to be operational all the time to have the latest information from the network. Also, depending on the type of information, OV will pull information and update its databases on an interval (configurable). So, for some application, these intervals may not reflect real-time changes, or the intervals should be changed manually. In case of Pscan, the port change information is updated every 5 minutes in OV which seems to be short enough for this specific application.

Requirements:

1. A Linux/Windows machine with IP connectivity to OV. This can be a Virtual Machine with 4GB of RAM and 40 to 60 GB of storage.
2. Linux/Windows machine being updated with the latest OS updates/patches.
3. Python 3.7 or higher being installed on the Linux/Windows machine.
4. A user with admin/sudo access on the Linux/Windows machine only to install python and required libraries. Script execution does not require admin/sudo access.
5. Read and Write permission on the script directory for the user running the script.
6. A user in OV with read/write access to OV
7. NTP service on OV and Linux/Windows machine needs to be active and synchronized by the same time source.
8. All the switches in the network should be synchronized by NTP to the same time source so that accurate information is provided to OV regarding port status change date/time. This is important since inaccurate date/time on the switch will result in script unnecessarily admin disabling the switch ports. This should not result in network down situations since the script only take action on the ports that are already operationally down. However, there is a risk in disabling the ports that are supposed to be active, but the switch has inaccurate time.

Running Pscan.

Pscan moslty uses standard Python libraries. There are couple of extra python libraries including “requests” and “cryptography” that needs to ne installed on the linux/windows machine. If there is a missing library on the machine, the script will stop execution with error which in that case we will need to install the missing library on the machine.

The tool connects to OV and downloads list of all devices that are discovered by OV. Please note that if you have some devices in your network that are not discovered by OV, they will not be detected by the tool. Once a list of devices is created, the tool will gradually query OV database and pull the port status information from the OV for every device. The information that is gathered for each port includes:

1. Switch name : Switch name which is configured by system command
2. Switch IP : Management IP of the switch that has been discovered by OV. If switch have multiple IP interfaces, only the interface discovered as management interface by OV will be considered.
3. Switch Location : Location of the switch which is configured by “system location” command
4. Port Number : Port number
5. Port Admin state : Admin state of the port
6. Port Operation State : Operational state of the port
7. Link Change date : Last time link status has changed from Up to Down or from Down to Up
8. Inactive days : Calculated number of days from the last link change date to the current date/time

Once this information is gathered, depending on the script execution mode (described below in the Settings) the script could either generate a report (if running in “Discover” mode) or it could Admin down the ports (if running in “Active” mode). Script will consider the following criterion to make a decision on Admin disabling the ports:

* The port should be in Admin Status Up
* The Port should be in Operation Status Down
* The Port should be inactive for the defined number of days (described below in settings)
* The port should belong to a switch that is in the execution domain of the script (Described below in settings)

If all the above conditions are met and the script is running in Active mode, the script will query OV to admin down the ports.

It is recommended to run the script in Discovery mode to get a snapshot of the network and determine the range and accuracy of the tool. Once this information is gathered and verified, the tools can be run in Active mode to actively admin down the ports.

The tool can be run in two different ways:

1. Interactive (Menu). To run the tool in interactive mode, use “python3 pscan.py”. This will present you with a menu with 3 options.
   1. [1] Settings : Used to configure some settings for the tool. Upon first execution of the script, a new file called “config.ini” is created in the same directory as script. This file includes all the settings and should not be deleted. The Setting menu includes:
      1. [1] Show Current Configurations : shows the current settings
      2. [2] Set OmniVista IP address : use to configure OV IP address. Required before first execution.
      3. [3] Set OmniVista Credentials : use to se OV credentials. This should be an OV user with read and write access.
      4. [4] Set Number of Inactivity Days : Set the number of days that the tool considers when making a decision on admin disabling the ports. The number of inactive days that is pulled from OV for a port should be equal or greater than this number for the script to act.
      5. [5] Set Script Execution Mode : use to set the script in Discover or Active mode. In Discover mode, the script will only gather information from the network and generate a report without taking any other actions. In active mode, the script will Admin down the ports that match the criteria.
      6. [6] Set Script Execution Domain by IP subnet : use to set one or multiple (separated with commas) IPv4 subnets that script will consider during execution. Default option is “0.0.0.0/0” which means that the script will scan and can act on the entire network that is discovered by OV. You can limit that by specifying one or multiple IPv4 subnets so that only the switches in these specific subnets are considered by script. Type “0.0.0.0/0” if you want the script to run on entire network. Please note that you would need to enter “Network/MASK” Address. For example: '192.168.1.0/24, 172.16.0.0/16'. If the network address you enterd is invalid, the script reject and ask for reentry. Supernetting/Subnetting with a correct subnet mask is supported.
      7. [0] Back to Main Menu : exit the setting menu and goes back to main menu.
   2. [2] Execute
      1. Execute the tool from the menu
   3. [3] Quit
      1. Quit the tool
2. Auto mode : To run the tool in auto mode (non interactive) use “python3 pscan.py -a”. Adding “-a” argument will run the tool automatically and skip the menus. This is appropriate mode when scheduling to run the script automatically using Cron jobs/Windows task schedular. For the script to work properly in auto mode, you should have already configured the settings mentioned above. So, normally you will run the script in Menu mode initially, configure the settings, test the execution and after that you can run the script automatically for next executions.

Logging:

The script will generate a log file after each execution whether running in Discover or in Active mode. This log file is created in the same directory as the script and named pscan\_date\_time.log. The log file contains all the information regarding discovery, actions and any petential execution errors. The maximum time interval for log creation is 1 minutes, meaning that if script is executed more than once in a minute, the logs will be in the same file timestamped with the same minute.