All the scripts can be found in the GitLab, in the Portmap project under the Netbox_dev subproject.

phase_2.py

Continuation of phase 1. Aims to integrate a full IPAM solution by associating the mac address for each child device on the port of an ONT to their respective IP address. The output should be a populated NetBox knowledge of which devices are connected to which ports on an ONT, as well as the mac and IP addresses of the connected child device.

A matching algorithm is used to compare the mac address to the IP of the discovered device and to its associated mac and ONT in NetBox.

NMAP has been configured to scan the different VLANs associated with the ONTs. The results of the NMAP scans are collected, which will provide the mac and IP addresses of the discovered devices on each VLAN. The matching algorithm used uses the mac addresses as a comparison point. The algorithm matches the IP addresses of the discovered devices with their associated mac addresses and ONTs in the NetBox database.

Phase 2 aims to enhance the functionality of the system by providing an IPAM solution that integrates with the existing infrastructure. This will enable the tracking and management of IP addresses associated with devices connected to ONTs, enriching the data available in the NetBox database.

ip_list.py

This code reads an input file (an demo data txt file) and parses through it. Only those with EPHY (or whatever is specified) as the interface are extracted and their IP addresses added to a new set, with the duplicates being removed. The initial 0 of the IP address is removed so it is read in proper format. Make sure that no duplicates exist, and then print out a list of valid IP pairs and its associated VLAN for a specified interface.

new_install.sh

A code to check if a cron job already exists. If there is a cron job already, the installation will be skipped and new cron jobs will not be created. All that will run is entering the site code and creating the .env file before moving onto the next steps. If no cron jobs exist already, the entire install.sh script will run and the cron jobs will be created.

Auvik/NetBox are very particular about overlapping or duplicate cron jobs, so be careful about this.

When running this new install, the cron schedule for the run.sh cron job must be specified in the command line as follows:

```
./install.sh “15 1 * * *”
```

*can add any time wanted, this would be scanned once a day at 1:15am.

Also be sure of the time zone, as Auvik terminal is typically in a different time zone, so add the cron job time to reflect this.

input_install.py

Another version of the install.sh script, where the user is prompted to input a specific cron schedule for the run.sh script. This is an improvement made to have flexibility with the input scheduled time. Multiple sites running at the same time causes issues in both the Auvik terminal and in NetBox. This way when a site is run on Auvik and cron jobs are created, the user can pick an unused time in order to avoid any errors.

ports-per-ont.py

The goal of this function is the have an output of the number of ACTIVE ports associated with each ONT. This would help know which ports were active earlier in the day if something goes wrong, there needs to be a written record of what was active before the issue occurred. This function would be added to the "utils.py" script in the AuvikScripts folder. In the getONT.py script, after the "utils.makeCSV()" line, "utils.countPortsPerONT()" would be added.

This function reads the portmapCSV.csv file that is already created after running getOT.py. This file gets read, and then the ONT IP addresses are grouped. The number of rows per group is recorded, and then a final output shows the ONT IP address and the port count for each one.

modified_ont.py & netbox_utils.py

The modified_ont.py script is the same as the original getONT.py, with a few added lines. The following changes associate mac addresses on each ONT port to an IP into the automated workflow. The final output should be a populated NetBox knowledge of which devices are connected to which ports on an ONT and the mac and IP addresses of the connected device. The added lines in this script match the mac addresses with IP addresses and update the csv data.

The netbox_utils.py script deals with the NetBox configuration. It reads the updated CSV data, then iterates over the lines to import the data to NetBox. The device is then found or created in NetBox, then the connected device is found or created. A new interface for the ONT port is created and one for the connected device. Finally, a connection is created between the interfaces. Some parameters in this code will need to be adjusted to match the NetBox setup.

With these changes, after running getONT.py:

-the network will be scanned for ONT devices and Telnet connections will be established

-the data will be parsed, and CSV files will be generated

-MAC address to IP address mapping will be performed for each VLAN, and the CSV data will be updated

-the ONT and CSV files are cleaned up, and the updated data will be imported to NetBox, associating devices with their respective ONT ports based on MAC and IP addresses

phase2_testing.py

Similar to ip_list.py, but the output is in a different format. The idea is the same, but the process is different. First, the file is parsed through and each row is checked to see if the interface is the specified one (EPHY in this case), and then if it matches then that row is added to an empty list. This list of rows with the specified interface is printed. Next, any duplicates are deleted. A new set is made to store unique IP addresses, and another list is created to store the duplicates to be deleted. The list is parsed through and all unique IP addresses are added to the set, and any duplicates are deleted. Finally, the 0 is removed from the beginning of the IP address from the original file so that it is in the correct format. A new file is created and is printed in a table format.

draft_phase2.py

Another write up for phase 2 of the Scripting Project. Similar to phase_2.py. This code establishes a telnet connection to the ONTs at a specific site and port map data is extracted. The data is parsed through, and a scanning method is used to determine the IP and MAC addresses. A list is created that will contain the port, MAC address and IP address. NetBox is then populated with the ONT data, updating the devices and interfaces with the corresponding MAC and IP addresses. This is another method to extract the necessary data and automatically populate NetBox, which is the final goal of the project.

*install.sh has had many different changes made to it. They all need to be tested in a real environment, and then the changes can all be combined together.