INSRUCTIONS FOR AUTO DISCOVERY AND NETWORK MAPPER FOR ZABBIX

Script developed using Python and Pyzabbix libraries

Abstract

This guide contains guides on how to use the codebase to manage scripts of auto discovery and network mapping when provisioning Zabbix instances on client environments.

Aman Thapa Magar

aman.yhbs@gmail.com

Table of Contents

Table of Figures

1 Pre-requisites:

Python, pip, curl, expect and pyzabbix package are required for Zabbix API to perform operations.

```
# Run the first Python script
echo "##################### Running AutoDiscovery Script ##################"
sudo python3 autodiscover.py
echo ""
echo "Auto Discovery Completed"

echo "Network Map Created. Please arrange the map to your suitable topology and
perform any other manual changes to your liking !!"

~
~
```

Figure 1: Main.sh file

1. The script is controlled by bash file **main.sh** which runs a python file named (**autodiscover.py**). This file has been upgraded and the network mapper has been combined into it as well so all the program is run from one script.
2. The **netmapfunc.py** script handles all necessary function to request calls from Zabbix API for desired hostgroup.
3. The client requirements have been added by automating most of the features and requires less manual intervention in the code, so now everything is prompted where user just needs to put correct set of answers to proceed with the script. Additionally, mapping of the templates of different cctv vendors installed within the installer has also been added in the action rules of the script.

2 Main Code

2.1 Autodiscover and Actions:

```
def authenticate_user(api_address, zabbix_user, zabbix_password):
    # Replace with your Zabbix server information
    zabbix_url = f'http://{api_address}/api_jsonrpc.php'

    # Attempt to authenticate user
    zabbix_api_url = f"http://{api_address}"
    zapi = ZabbixAPI(zabbix_api_url)
    try:
        zapi.login(zabbix_user, zabbix_password)
        return zapi  # Return Zabbix API object upon successful authentication
    except Exception as e:

        print("Authentication failed, Please Try Again")
        print()
        return None  # Return None if authentication fails


# Running the command to get the IP address using hostname -I
result = subprocess.run(["hostname", "-I"], capture_output=True, text=True)
if result.returncode == 0:
    api_address = result.stdout.strip().split()[0]
print('Enter Zasya Username:')
zabbix_user = input('')
print('Enter Zasya Password:')
zabbix_password = getpass.getpass('')

# Loop until authentication is successful
while True:
    zapi = authenticate_user(api_address, zabbix_user, zabbix_password)
    if zapi:
        break
    else:
        result = subprocess.run(["hostname", "-I"], capture_output=True, text=True)
        if result.returncode == 0:
            api_address = result.stdout.strip().split()[0]
        print('Enter Zasya Username:')
        zabbix_user = input('')
        print('Enter Zasya Password:')
        zabbix_password = getpass.getpass('')

zabbix_url = f'http://{api_address}/api_jsonrpc.php'
```

Figure 2: Added auth mechanisms when running script

Previously we had to set **Zabbix_user, Zabbix_password and Zabbix_url** by opening the Autodiscover.py file and changing values, as per client, I have removed the manual part where you need to go to the file and edit instead when running the script the user is automatically asked of the Zabbix username and password, the IP address is automatically taken from the host using subprocess linux command (**hostname -I**).

2.2 Step 1: Run main.sh file:

```
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# ./main.sh
###################### Running AutoDiscovery Script ##################
Enter Zasya Username:
Admin
Enter Zasya Password:


Authentication failed, Please Try Again

Enter Zasya Username:
```

Figure 3: Failed authentication, Try Again feature

If the username and password don't match as the Zabbix user created during installation, then it will deny access to login and update any API push and re asks for correct username and password. This ensure users are not exited out of the script or program and they don't have to rerun the script repeatedly.

2.3 Step 2: Enter desired network range and interval check:

```
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# ./main.sh
################### Running AutoDiscovery Script #################
Enter Zasya Username:
Admin
Enter Zasya Password:

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   199    0    68  100   131    565   1088 --:--:-- --:--:-- --:--:--  1658
Authentication successful. Auth token: cde22b288eaa6492280669e768fb00ec

Host group 'IP Camera' created with ID: 25

Enter network range you want to discover, you can use commas to seperate IP address or IP ranges, for eg 192.168.0.1,192.168.1.8 or 192.168.1.0-254,192.168.2.0-254:
192.168.1.1, 192.168.1.3
Enter interval check for discovery to run, for eg 1m or 5s or 1h:
10s
```

Figure 4: Enter new ip range and delay for discovery rule

If you are adding devices for the first time, then you will be prompted with just question highlighted in red in the above picture. The input format for adding ip ranges should be in this format (i.e., **192.168.0.1,192.168.1.1 or 192.168.0.1-254,192.168.1.1-254**) and for adding interval checks it should be in this format (i.e., **5s, 1m, 1h where s stands for seconds, m for minutes and h for hours**). It is to be noted that it will take atleast 2 min to fetch the data properly from the discovery rules and actions created which can be observed in the pictures below:

2.3.1 Step 2: Results in the Web UI

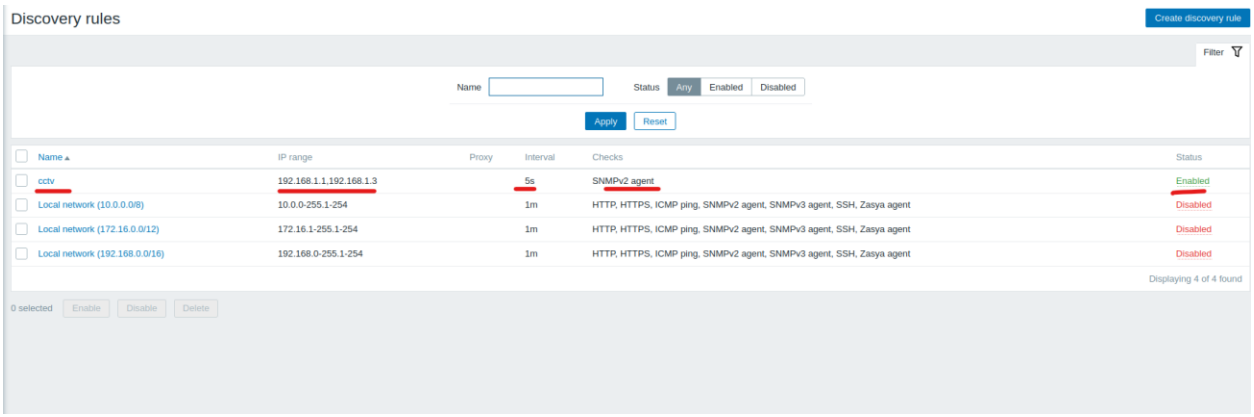


Figure 5: Evidence of discovery rule created after runnin script

The above script after running creates **cctv** discovery rule which can be verified in Discovery Rules navbar in web UI.

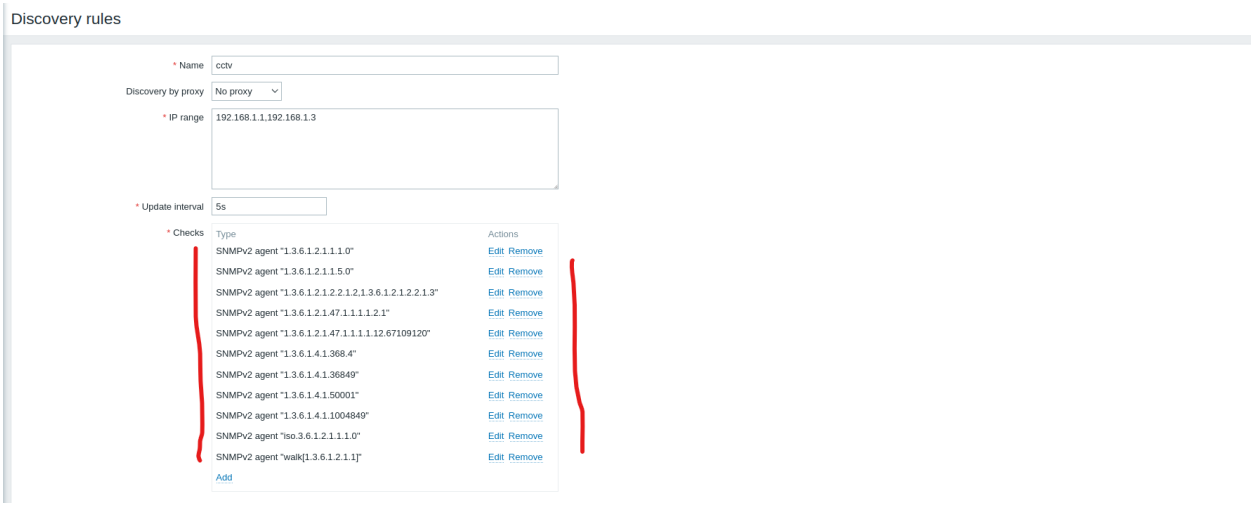


Figure 6: Evidence of snmp checks added as per client documentation

As per agreed with the customer, the OID checks have been added which was said to be universal for all devices as per client so, no need to add OID checks, should they require to add then they just need to edit the **configure-checks.txt** file located inside netmapper directory and just change values as per their liking.

```
iso.3.6.1.2.1.1.1.0 public 161
1.3.6.1.2.1.1.1.0 public 161
1.3.6.1.2.1.1.5.0 public 161
1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3 public 161
1.3.6.1.2.1.47.1.1.1.1.2.1 public 161
1.3.6.1.2.1.47.1.1.1.1.12.67109120 public 161
walk[1.3.6.1.2.1.1] public 161
1.3.6.1.4.1.50001 public 161
1.3.6.1.4.1.36849 public 161
1.3.6.1.4.1.1004849 public 161
1.3.6.1.4.1.368.4 public 161
```

Figure 7: Configure-checks.txt file contents located inside netmapper directory

Just make sure you keep the columns intact and not modify it as it needs to be exact if we add more columns than the script can take it will cause problems so whenever we are adding any additional OID place it in the file as follows:

[oid] [community string] [port]

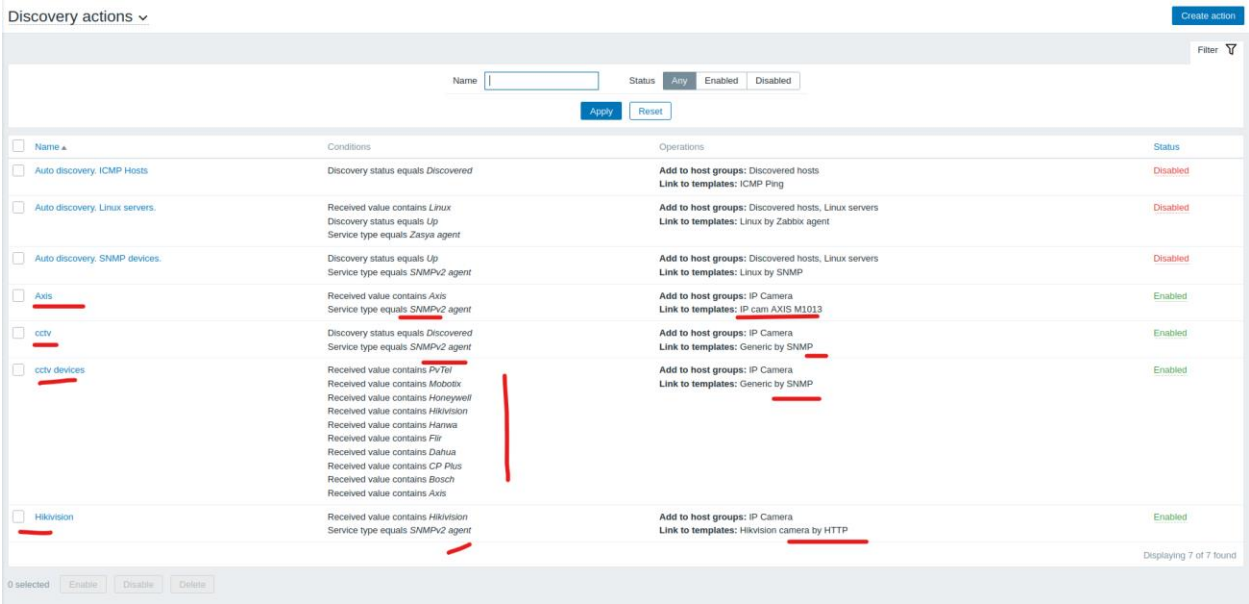


Figure 8: Evidence of discovery actions created after running script has client requirements of mapping to diffrent camera templates and string based checking

As per agreed with the client, I have added the following discovery action by mapping to the templates as shown by the client. However, I could only find Axis and Hikivision Template thou the client did say they had all the cctv devices template. Whatever I could find I have mapped it, rest will depend upon manual addition. So for now, as per the image above highlighted In red, it does fetch data for graphing based on the values such as PvTel, Mobotix, Hanwa etc string if they find it when discovering devices.

2.4 Step 3: Adding hostgroup name for Network Mapper Script to Map Hosts:

After the autodiscovery action and rules are created the zasya services are restarted to take changes to effect. then after some wait time of about 2 min to fetch latest data, you will be prompted to add your hostgroup in the terminal, you can head to your Zabbix web dashboard navigate to sidebar >> data collection >> Hostgroup >> There you will see a name called IP Camera which was created using the below static name already defined in the code.

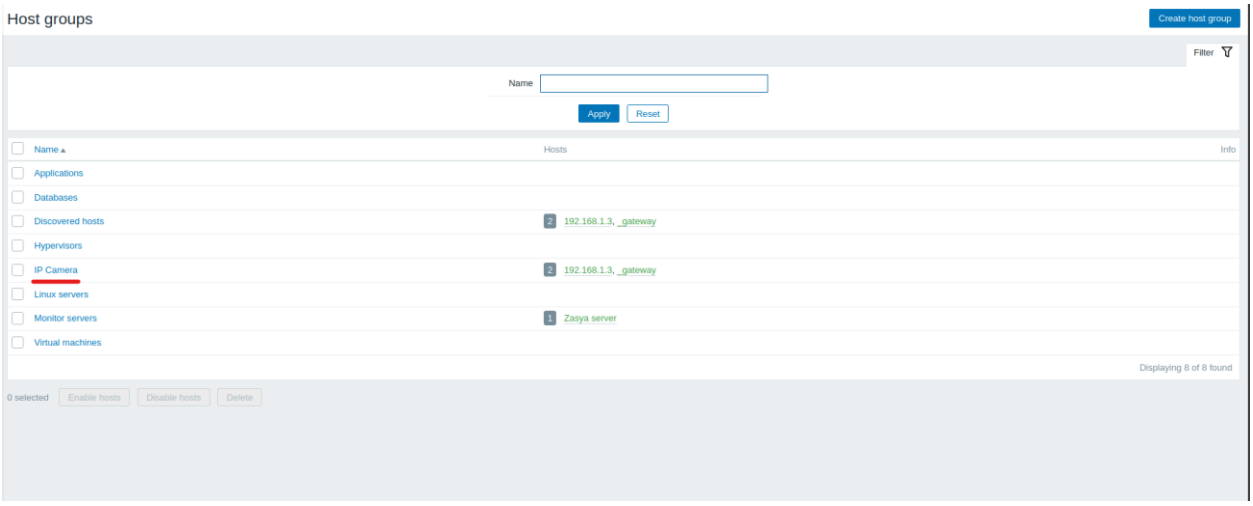


Figure 9: Evidence of hostgroup with name IP Camera added via script

```
# Create or get the host group named "cctv"
hostgroup_name = "IP Camera"
hostgroup_id = create_or_get_hostgroup(zapi, hostgroup_name)
```

Figure 10: Picture of the code in autodiscover.py where you can set your hostname to your liking

But this can be changed to your liking in the **autodiscover.py**.

After entering desired hostname, the script will then run the network mapper script for the desired hostgroup and starts mapping them in the map. The coordinates are also randomised in the map so that the devices are evenly distributed around the map space. Please note that you may need to arrange it manually.

```
############## Starting Network Auto Map Configurator #############


Enter the hostgroup for which you want to make a map, format, exact syntax needed
IP Camera

Map 'IP Camera' does not exist. Creating...

Custom icon 'cctv_(64)' already exists with ID: 188. Skipping the upload.

Randomizing coordinates for new host 10615: X=710, Y=444
Randomizing coordinates for new host 10616: X=702, Y=771

Map 'IP Camera' created successfully.

Auto Discovery Completed

Network Map Created. Please arrange the map to your suitable topology and perform any other manual changes to your liking !!
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# S
```

Figure 11: Network Mapper Script Running to map devices.

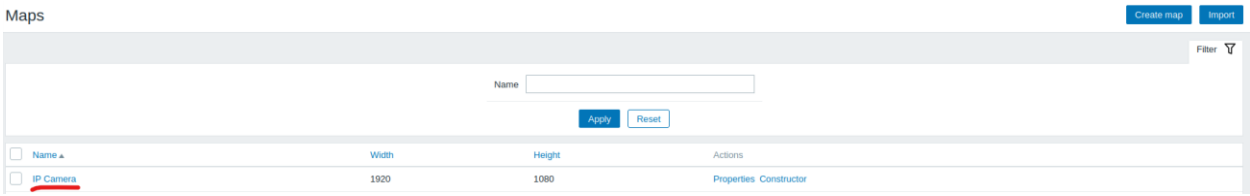


Figure 12: Map added in web UI

In the above picture highlighted in red, you can see map for hostgroup IP Camera has been created as per the input.

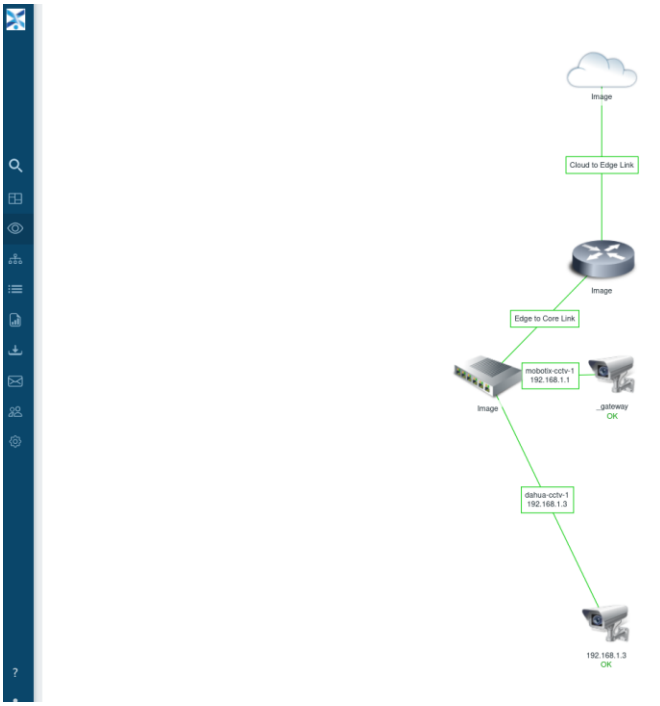


Figure 13: View of Map after script is executed as it shows the hosts I added via input

The diagram will look something like this, thou you may require to rearrange this map.

(i.e. one thing I noticed why your script of Autodiscover was not working was you were using a very large /16 pool to determine the ip address of the devices. The problem of Zabbix or one of drawbacks is it goes by sending hello packets one by one to each ip starting from the range till it reaches the desired Ip range block for example: if it traverses using 192.168.0.0/16 and your desired IP range to be scanned is 192.168.10.0/24 then, it will start scanning and sending one packet to each IP one at a time like 192.168.0.1, 192.168.0.2……etc and so on. Which is a not a reliable source to stay with. Hence, to eradicate the issue, if the ip addresses range are known that we can eliminate the process of scanning each individual network range and automatically add devices.)

NOTE: what I am talking about can be found using tail -f /var/log/Zabbix/Zabbix_server.log file but you will need to enable the /16 pool and restart the server to see the errors.

2.5 Step 4: Adding more IP ranges, updating delays and SNMP OID checks.

2.5.1 Adding IP ranges and Updating Delays

The process for updating or adding parameters have been made simple, users don't need to go inside the program code to change variables they only require rerunning the main.sh script again like below:

```
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# ./main.sh
##################### Running AutoDiscovery Script ##################
Enter Zasya Username:
Admin
Enter Zasya Password:

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   199    0    68  100   131    660   1271 --:--:-- --:--:-- --:--:--  1950
Authentication successful. Auth token: 3f3ab1cd21a33cb138b456de941e3dc1

Host group 'IP Camera' already exists with ID: 32
Enter additonal network range you want to discover, you can use commas to seperate IP address or IP ranges, for eg 192.168.0.1,192.168.1.8 or 192.168.1.0-254,192.168.2.0-254:
```

Figure 14: Script asking for additional IP range and new value for delay after detects the discovery rule exists

As shown in the above picture, you will be prompted for username and password of Zabbix for security purposes, and now instead of new you will be prompted with a different question for input, which is to add additional network ranges, in this section just add more (**i.e., we already had added 192.168.1.1, 192.168.1.3 now we will add 192.168.168.2.1, 192.168.3.1, 192.168.4.1**) and update our delay to **5s** as same as previous for testing as shown in the figure below:

```
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# ./main.sh
##################### Running AutoDiscovery Script ##################
Enter Zasya Username:
Admin
Enter Zasya Password:

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   199    0    68  100   131    660   1271 --:--:-- --:--:-- --:--:--  1950
Authentication successful. Auth token: 3f3ab1cd21a33cb138b456de941e3dc1

Host group 'IP Camera' already exists with ID: 32
Enter additonal network range you want to discover, you can use commas to seperate IP address or IP ranges, for eg 192.168.0.1,192.168.1.8 or 192.168.1.0-254,192.168.2.0-254:
192.168.2.1,192.168.3.1,192.168.4.1
Enter updated interval check for discovery to run, for eg 1m or 5s or 1h:
5s
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1435    0    53  100  1382    699  18237 --:--:-- --:--:-- --:--:-- 19133

Discovery rule updated for ID: 16 with new values.
Discovery action 'cctv' already exists.
Discovery action 'cctv devices' already exists.
Discovery action 'Hikivision' already exists.
Discovery action 'Axis' already exists.

############## Restarting Zasya Services ##################
Zasya Service restarted succesfully !
############## Starting Network Auto Map Configurator #############
```

Figure 15: New IP address and delay pushed to the server

After the services are restarted, you can observe the updated discovery rule by navigating to the UI.

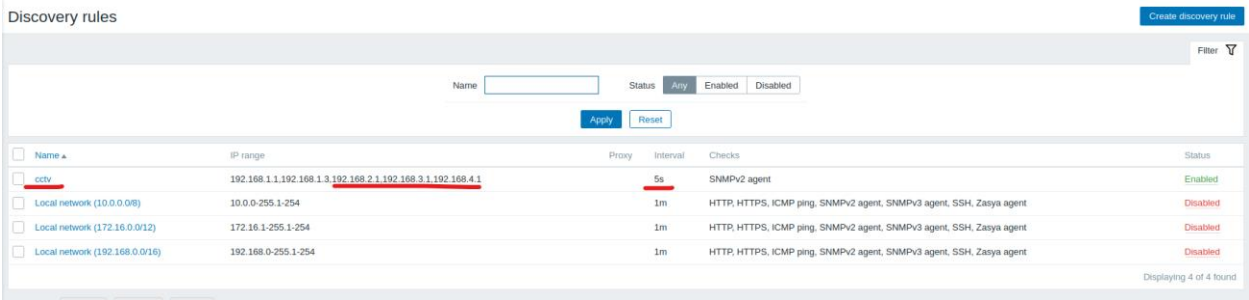
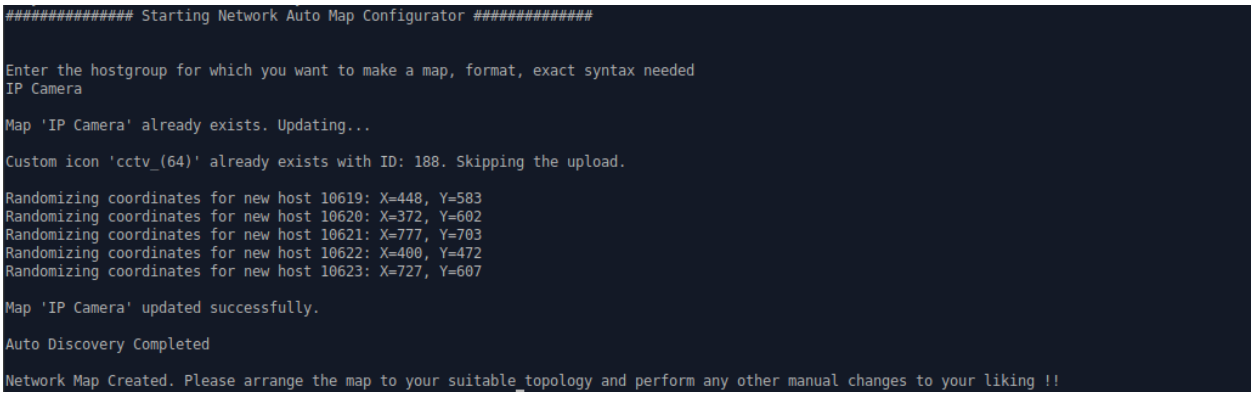


Figure 16: Evidence of IP address being updated with updated delay

Here, in the above picture you can see the new ip ranges and delays have been added as given input.



```
############## Starting Network Auto Map Configurator #############


Enter the hostgroup for which you want to make a map, format, exact syntax needed
IP Camera

Map 'IP Camera' already exists. Updating...

Custom icon 'cctv_(64)' already exists with ID: 188. Skipping the upload.

Randomizing coordinates for new host 10619: X=448, Y=583
Randomizing coordinates for new host 10620: X=372, Y=602
Randomizing coordinates for new host 10621: X=777, Y=703
Randomizing coordinates for new host 10622: X=400, Y=472
Randomizing coordinates for new host 10623: X=727, Y=607

Map 'IP Camera' updated successfully.

Auto Discovery Completed

Network Map Created. Please arrange the map to your suitable topology and perform any other manual changes to your liking !!
```

Figure 17: Script detecting newly added ip range and updating maps

After the services have been restarted and the desired hostgroup has been added, the newly added devices will also be created on the map automatically.

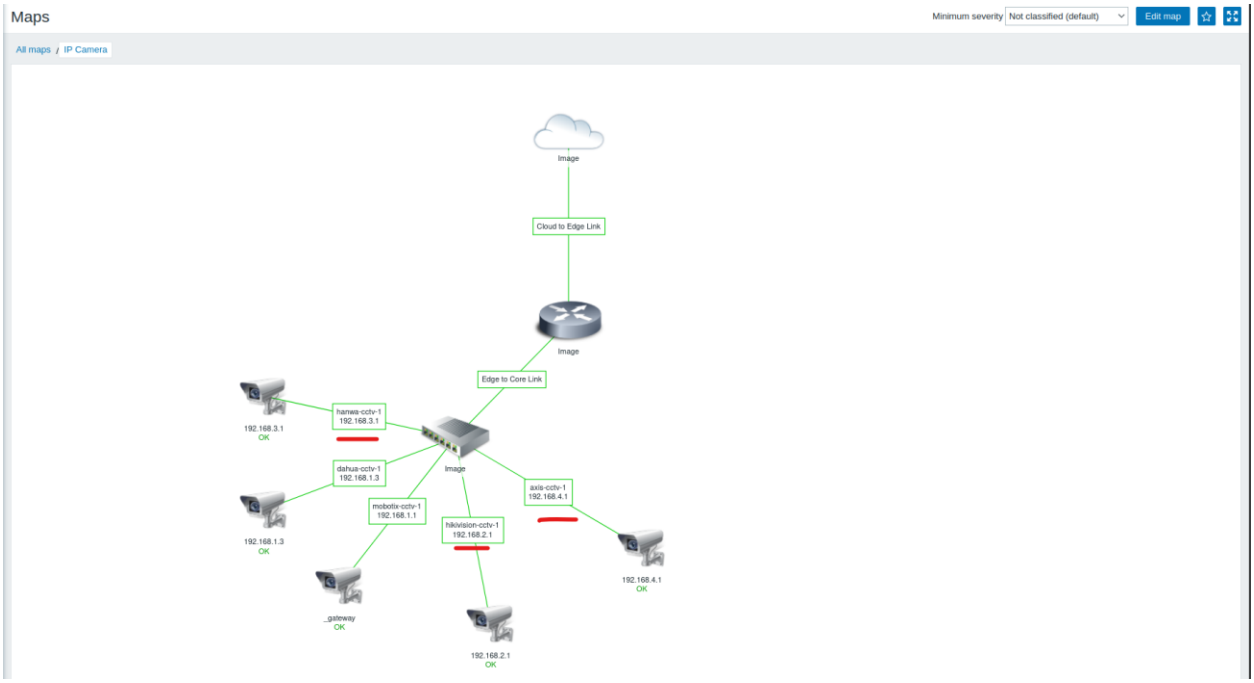


Figure 18: Evidence of newly added ip range being reflected highlighted in red

2.5.2 Updating OIDs, SNMP Community String

As mentioned earlier, you don't require manual intervention in the code. Instead, you just need to update configure-checks.txt file and the desired output will be served in the Web UI. For Example, right now I have configured the community strings to be "**public**" for all host as shown in the figure below:

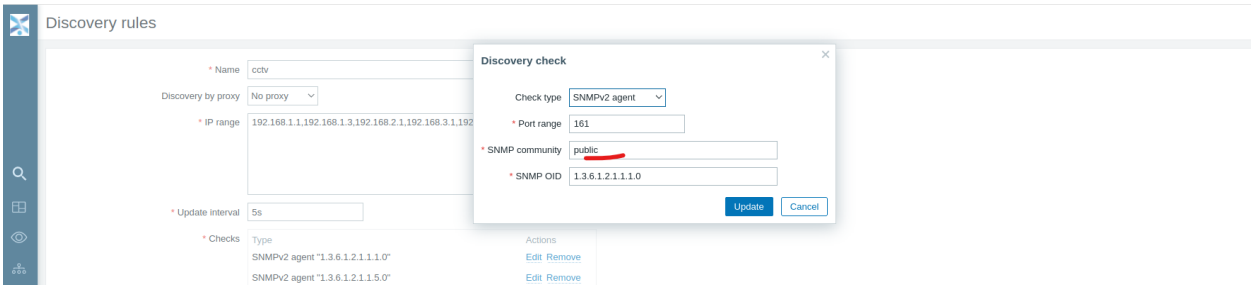


Figure 19: UI showing community string as public

Now if I want to change the community string to different name let's say to "zasya", then I would do the following:

Go to your terminal and to your netmapper directory. Open configure-checks.txt with your favourite editor (i.e., vi or nano) and edit the following as follows:

sudo vi configure-checks.txt or **sudo nano configure-checks.txt**

Figure 20: Using vi editor to perfom find and replace from public to zasya

So by issuing the command since im using vi editor since it is easier for me to use shortcuts then nano editor. Im issuing the command **"%s/public/zasya/g"** to find and replace all match cases of public to zasya. (i.e., **nano must have different shortcut to do find and replace**). This will result to the below image:

Figure 21: Values substituted from public to zasya

The lines have been substituted from public to zasya for the community string, now we rerun the script main.sh again to adhere changes.

```
root@Zabbix:/home/aman/Desktop/ZasyaMonitor-main/netmapper# ./main.sh
#################### Running AutoDiscovery Script #################
Enter Zasya Username:
Admin
Enter Zasya Password:

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   199    0    68  100   131    509    980 --:--:-- --:--:-- --:--:--  1496
Authentication successful. Auth token: 18cd4e9426fb830906da1ae4dfa33314

Host group 'IP Camera' already exists with ID: 32
Enter additonal network range you want to discover, you can use commas to seperate IP address or IP ranges, for eg 192.168.0.1,192.168.1.8 or 192.168.1.0-254,192.168.2.0-254:

Enter updated interval check for discovery to run, for eg 1m or 5s or 1h:

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1424    0    53  100  1371    819  21204 --:--:-- --:--:-- --:--:-- 22250

Discovery rule updated for ID: 16 with new values.
Discovery action 'cctv' already exists.
Discovery action 'cctv devices' already exists.
Discovery action 'Hikivision' already exists.
Discovery action 'Axis' already exists.
```

Figure 22: Rerunning the script again to reflect new community string and sending empty values in ip range and delay

You might have notice after entering my zasya username and password, I have not added anything in the network range and delay, this is because im just editing the community string and I don't have anything to add or update in my discovery rule. So, if you don't have anything to add or update and youre just updating you snmp oids and community strings just press enter by leaving the field empty. Now, you can go to the web UI to see the changes.

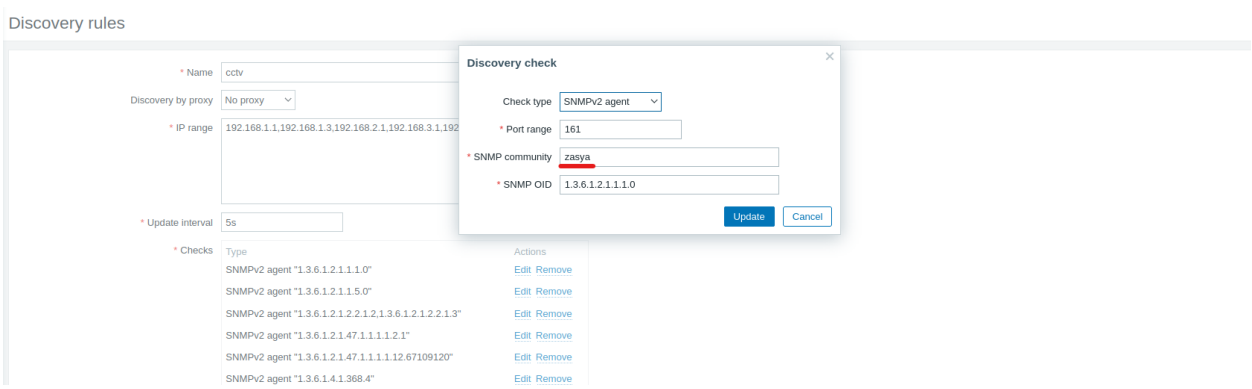


Figure 23: Evidence of community string has been updated with zasya from public

As you can see in the picture, after our configure-checks.txt file was updated from **public** to **zasya**, the server has been reflected with a new community string. Likewise, you can add more OID's you can edit one OID or multiple sets of OIDs, add delete etc and it will reflect in the discovery rule after running the main.sh script.

That's it. Happy Exploring !!!