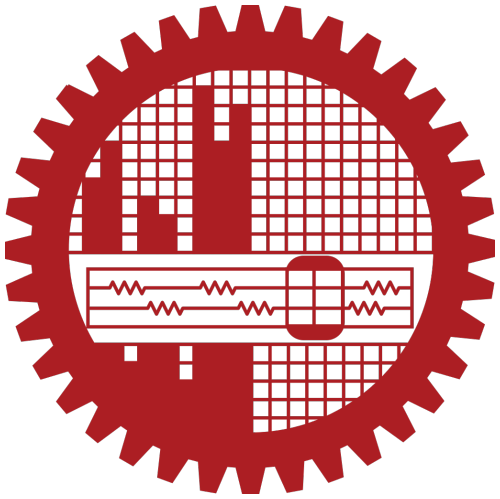
CSE 406
Term Project:
Port Scanner Tool



Md. Shanjinur Islam
Level 4 Term 1

Student ID: 1505066

Group: 03

Tool Introduction

I have implemented **Port Scanner Tool** using Python Programming Language. To run it, you will need only one file which is **main.py.** I have used positional and optional arguments for better utilization of the tool. In the following section, I will provide each features that can be done with my Port Scanner Tool.

Steps of using the tool

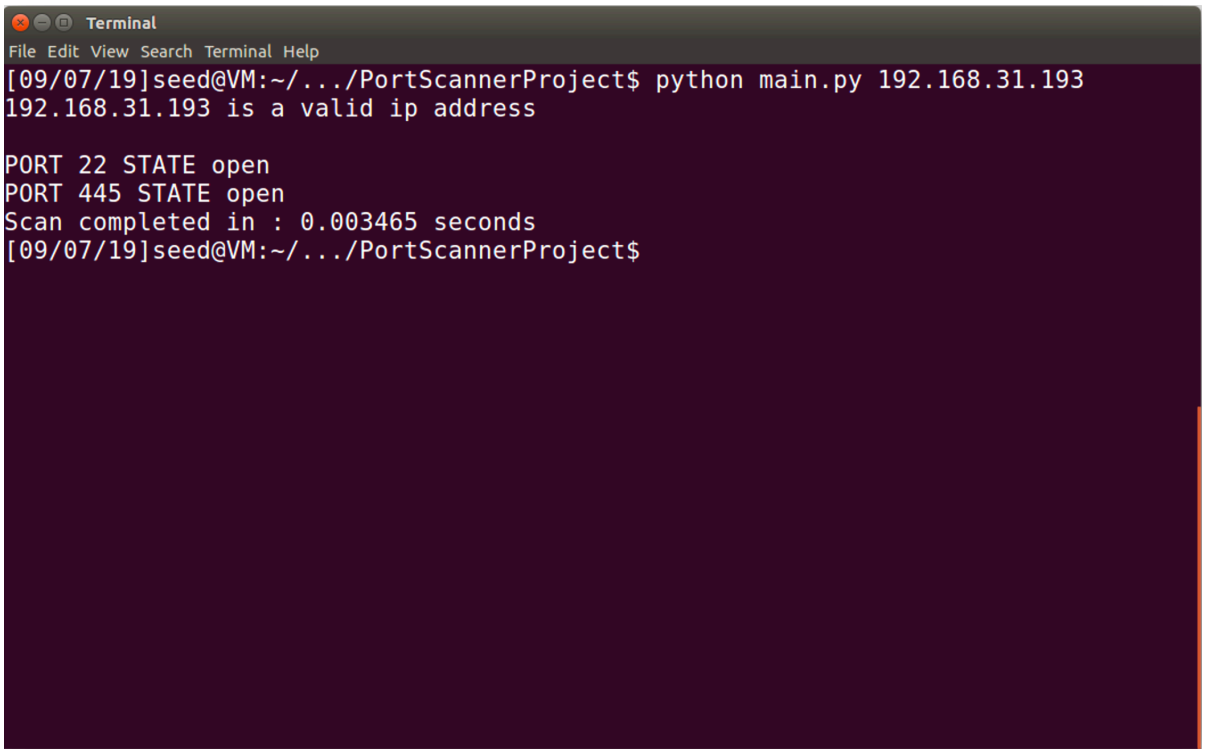
The tool is very easy to operate. The basic command is to:

python main.py {ip_address/url}

When no other optional arguments is provided it basically checks for open ports in usual Common port which are 1, 5, 7, 18, 20, 21, 22, 23, 25, 29, 37, 42, 43, 49, 53, 69, 70, 79, 80, 103, 108, 109, 110, 115, 118,119, 137, 139, 143, 150, 156, 161, 179, 190, 194, 197, 389, 396, 443, 444, 445, 458, 546, 547, 563, 569, 1080

For IP Address :

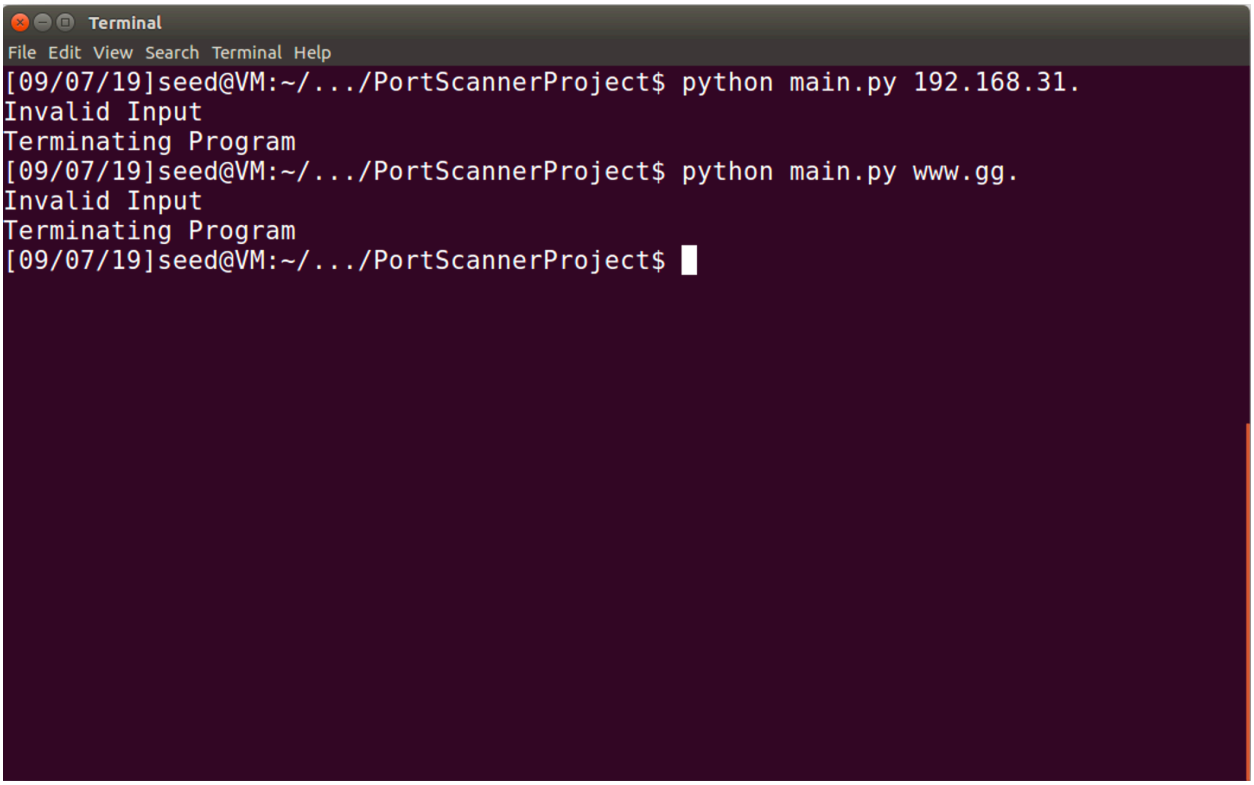
python main.py 192.168.31.193 (IP Address of a PC in my network)



Here, the positional argument I used is target IP address or URL. There is a validity checker whether the provided IP or URL is correct or not.

Here I provided two inputs where in first case invalid url and in second case invalid url.

In both cases it returns invalid input and terminates the program.



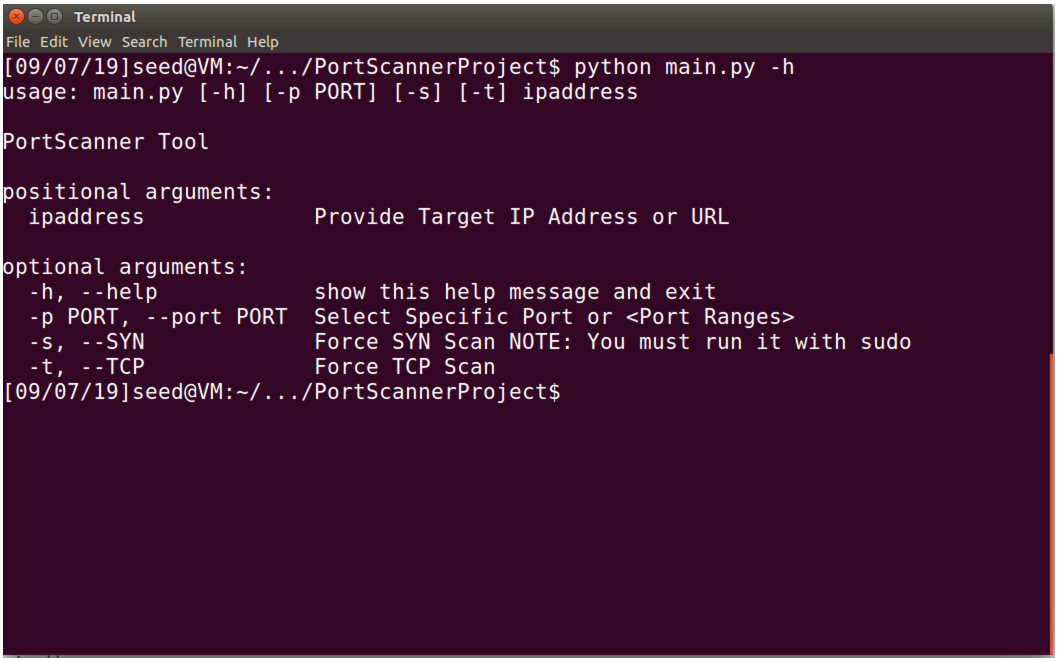
Scan Types:

Here I implemented two versions of Port Scanning as I mentioned in design report. They are:

- TCP Scanning
- SYN Scanning

For better user friendly experience I have provided a **help** screen to understand how to provide inputs.

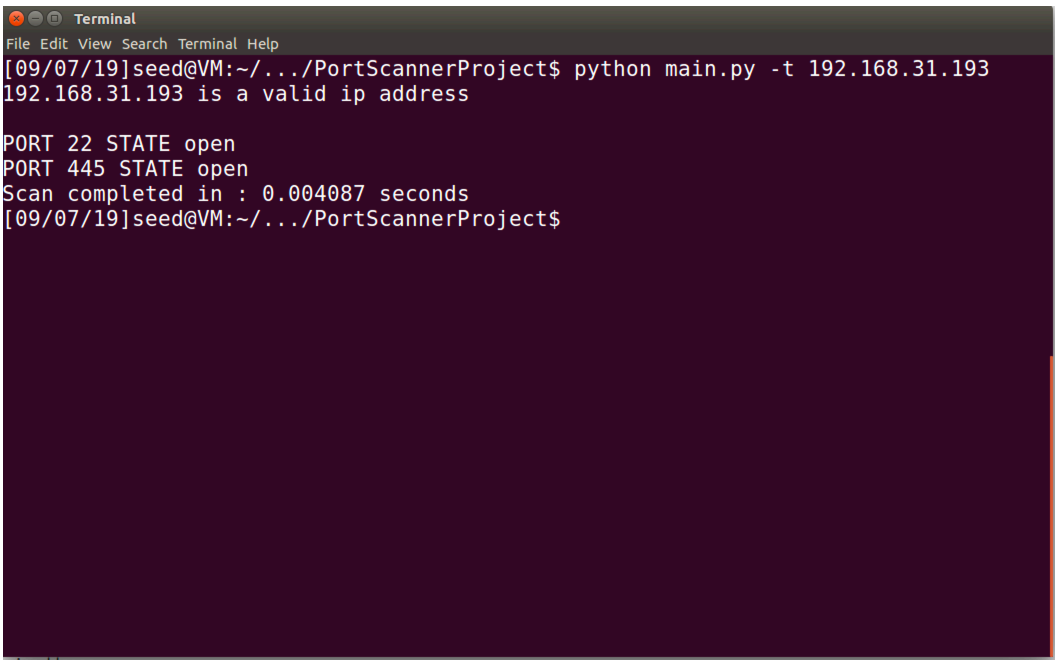
In terminal write : **python main.py -h**



TCPScan

Although default Scan type is set TCP Scanning but you can force it by following the command.

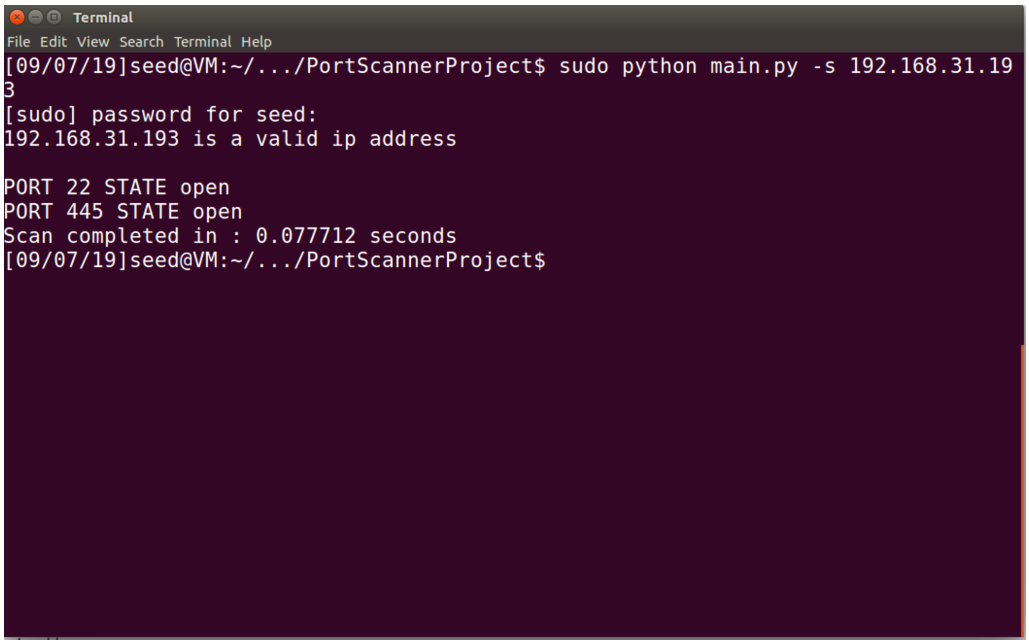
In terminal write : **python main.py -t 192.168.31.193**



SYN Scan

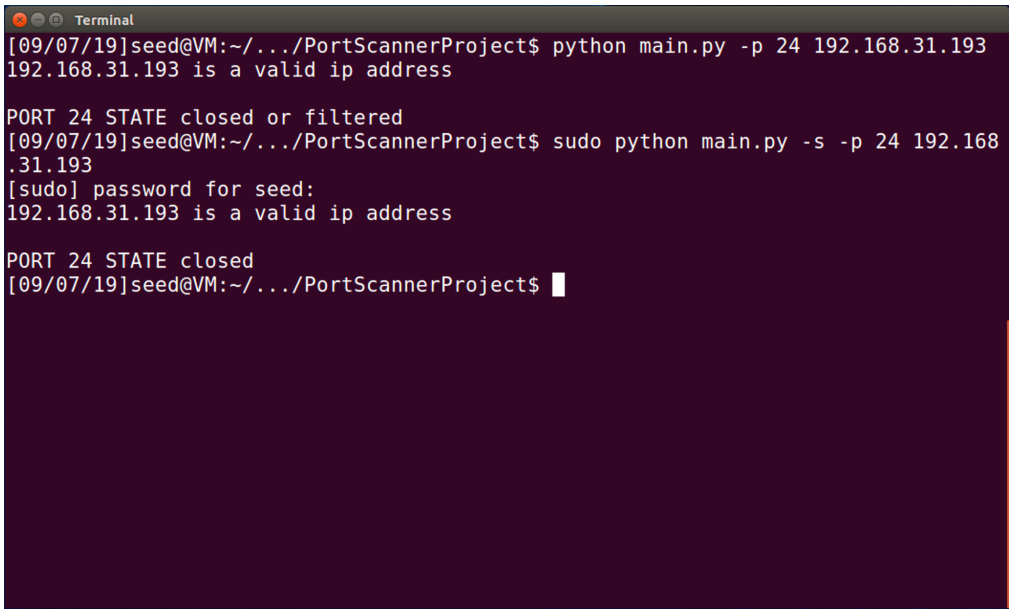
Although default Scan type is set TCP Scanning but you can force it by following the command.

In terminal write : **sudo python main.py -s 192.168.31.193**



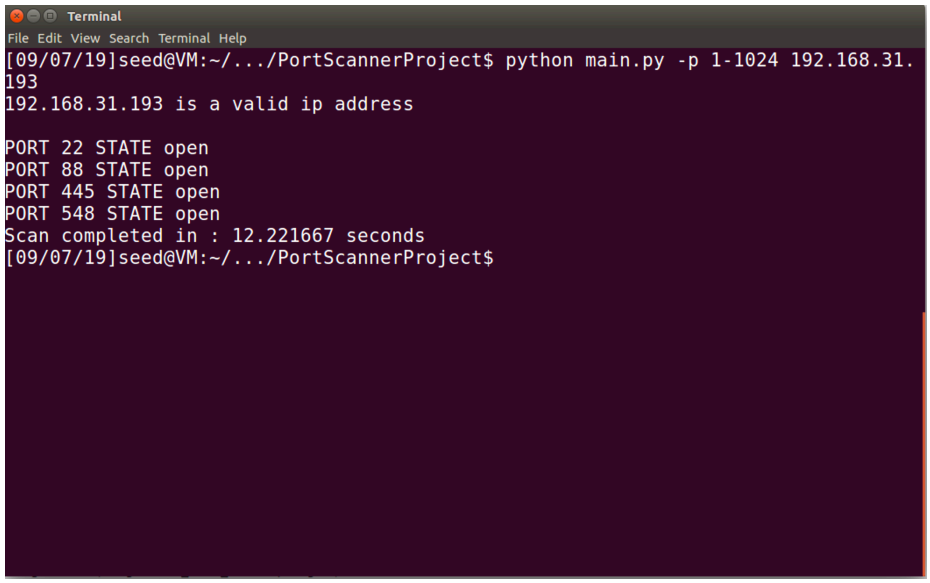
Specific Port or Range of Ports Scan

To scan a specific port write **python main.py -p 22 192.168.31.193.**
It will show whether the port is opened or closed.



To scan range of ports write **python main.py -p 1-1024 192.168.31.193**

Here 1-1024 means range from 1-1024

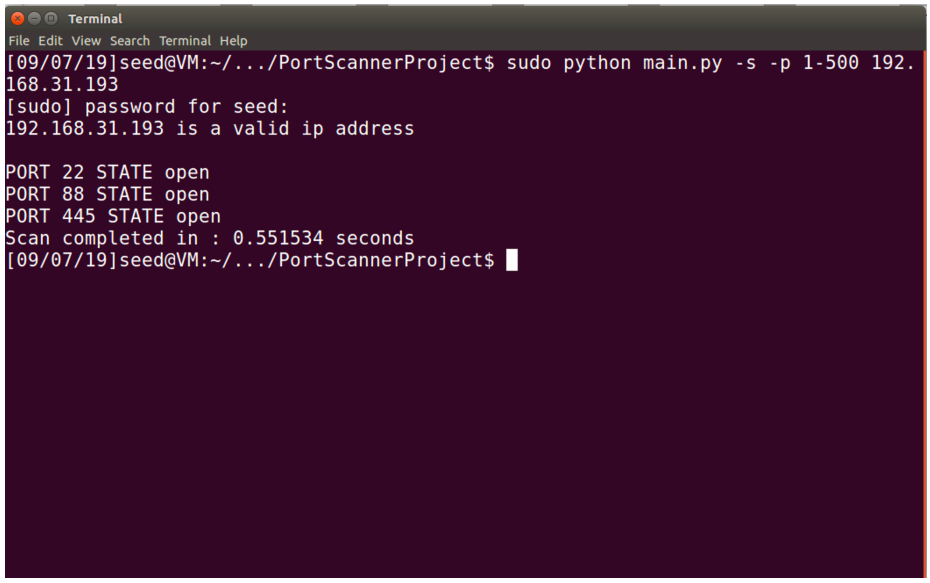


Using multiple options

You can use multiple options with a single command. For example:

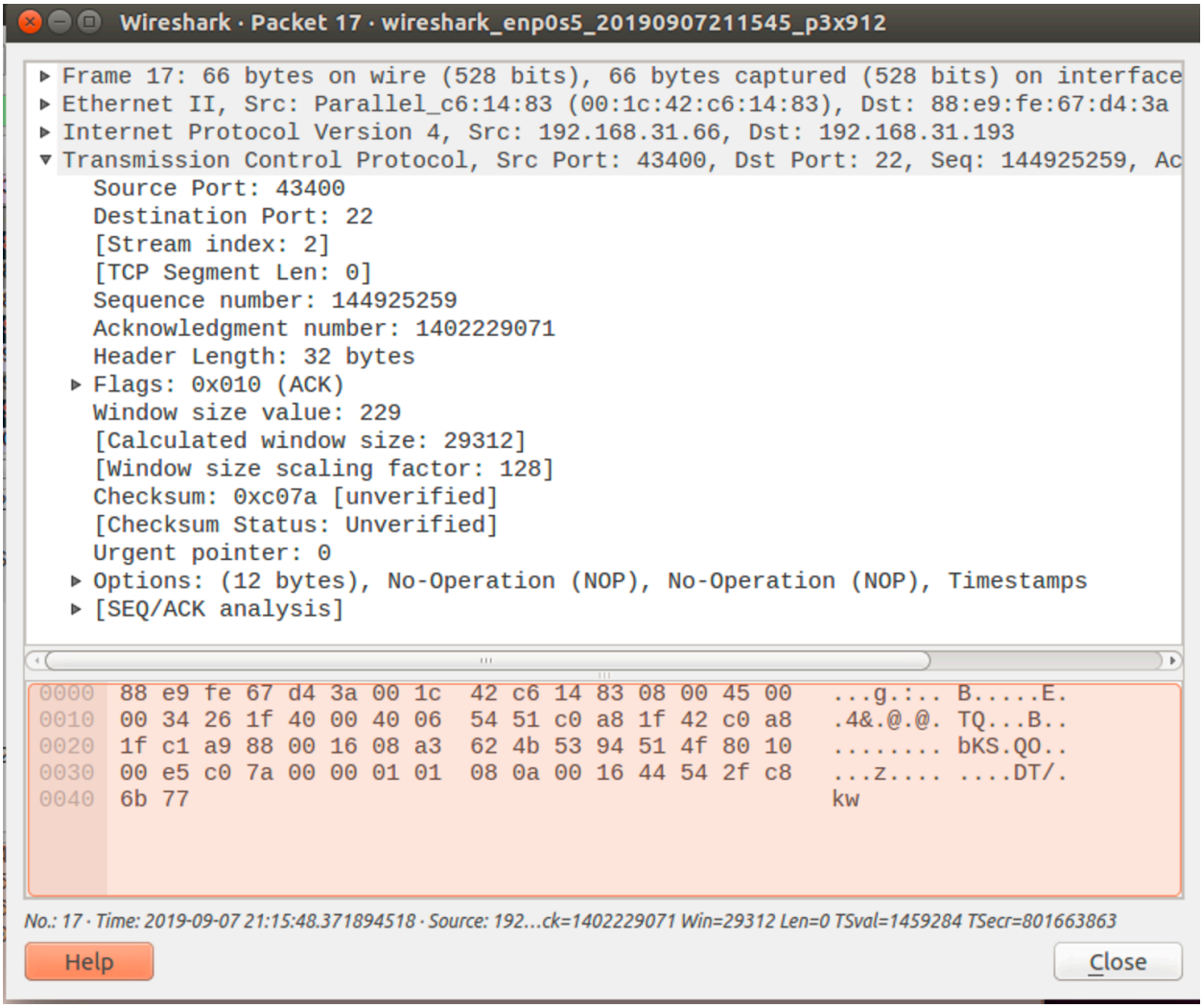
You want to do SYN Scanning over a range of ip from 1-500. So to perform this scan you have to write in terminal:

sudo python main.py -s -p 1-500 192.168.31.193



Wireshark Observations

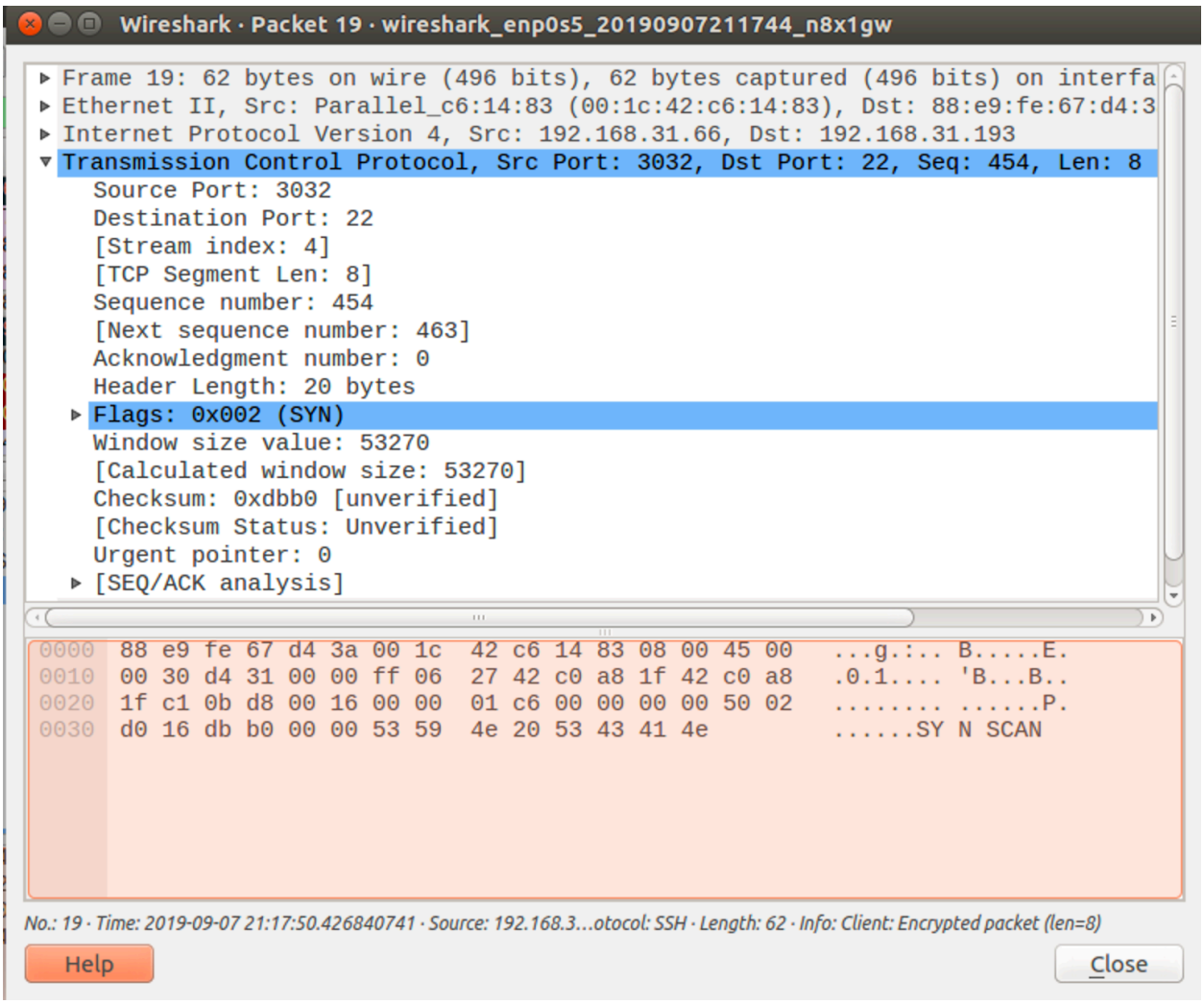
For a single port TCP scan , **python main.py -p 22 192.168.31.193,** Wireshark observation is given below:



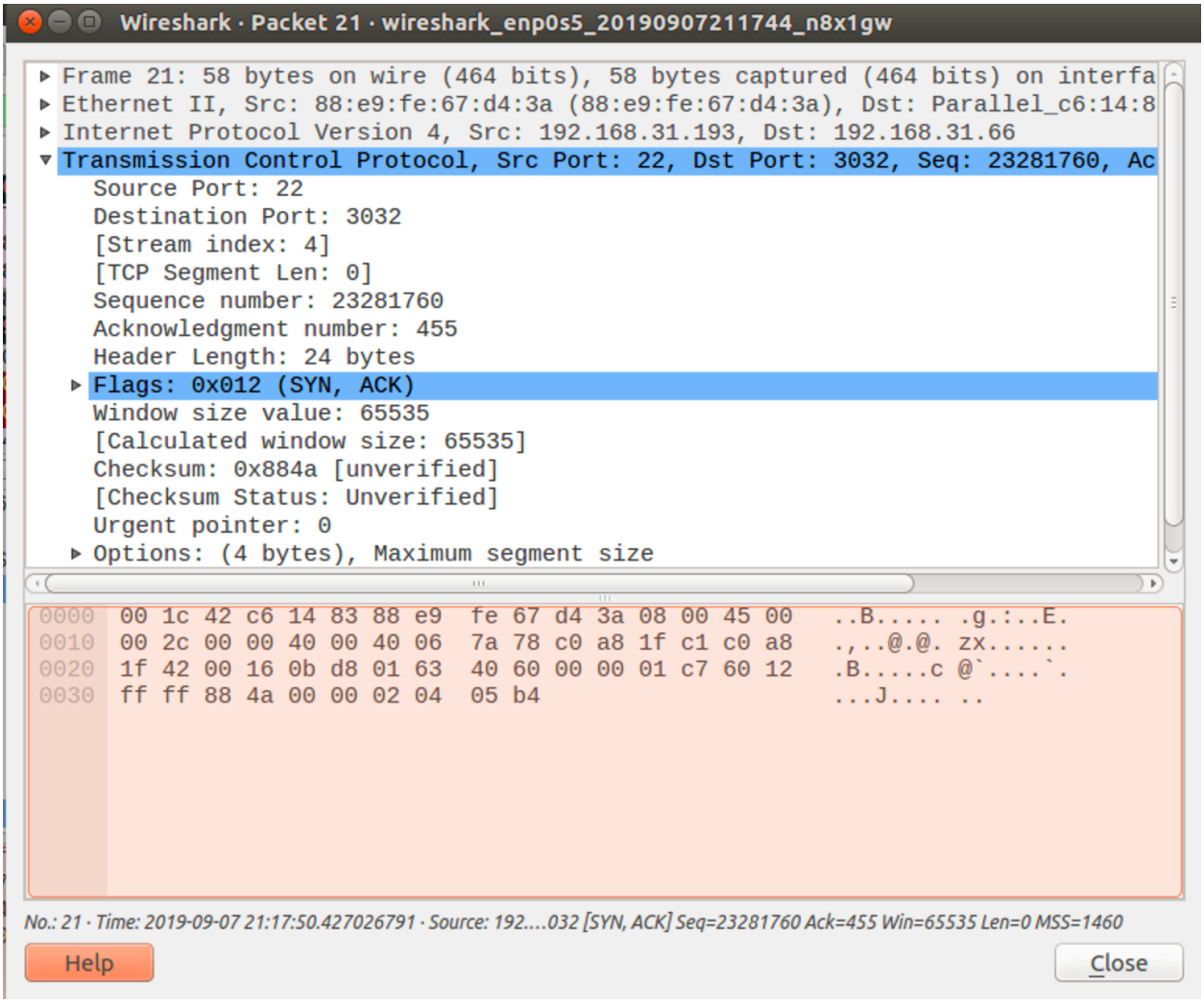
For a single port SYN scan , **sudo python main.py -p 22 192.168.31.193,** Wireshark observation is given below:

SYN Scan is a three step process.

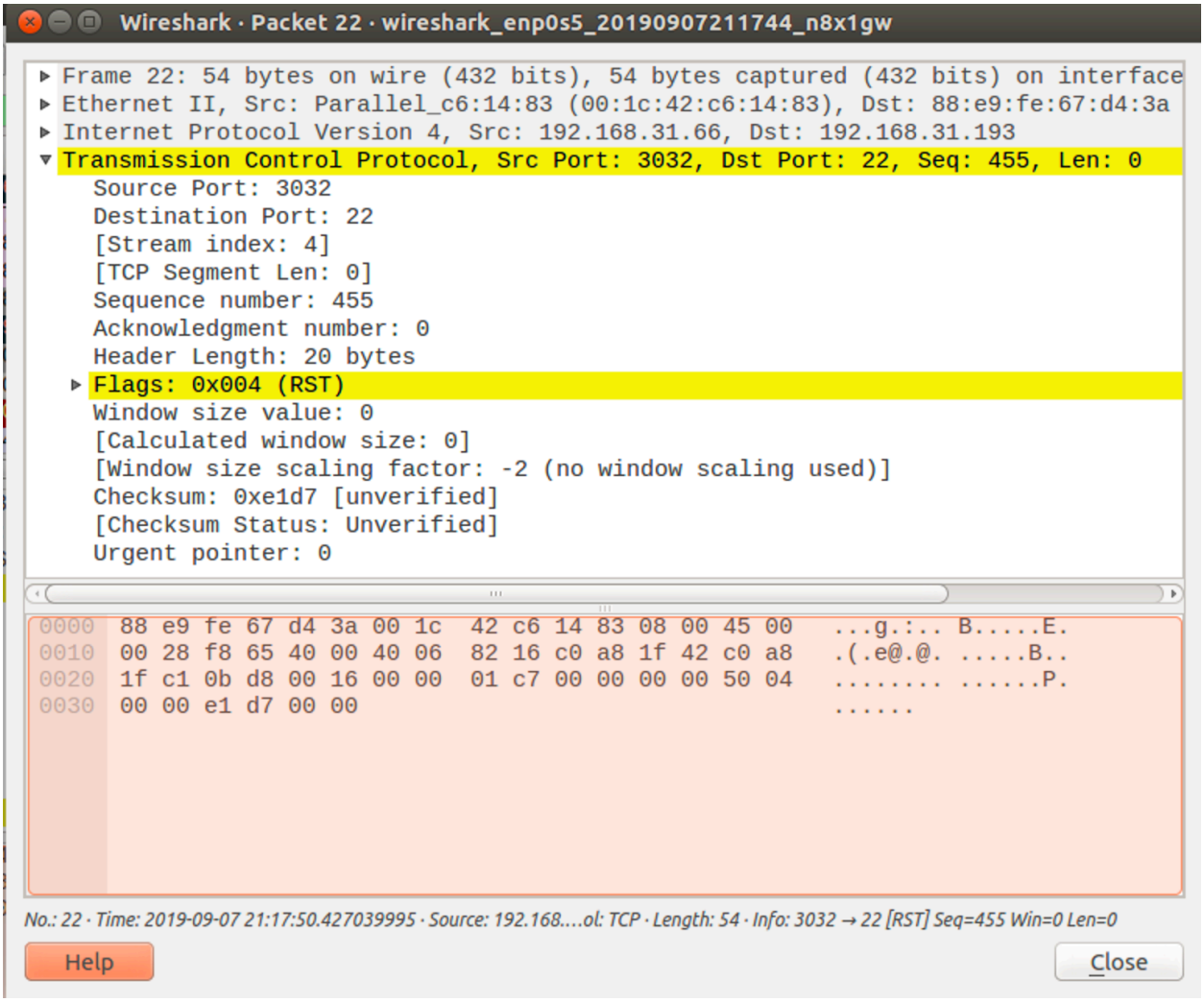
First Step is send a packet with SYN flag on.



Second Step is reception of a packet from destination with SYN and ACK flag on.

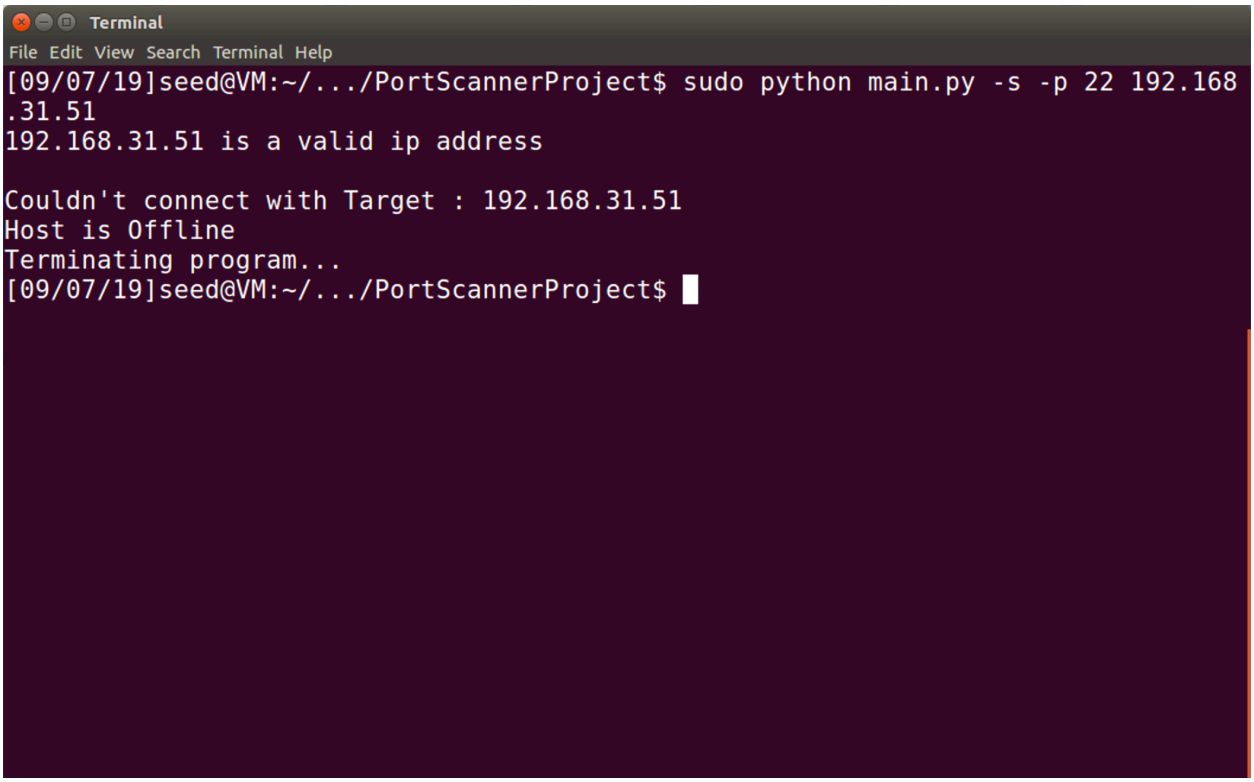


Third and Final Step is send a packet with RST flag on. The Screenshot is provided in next Page



Extra Features

If host is not connected to the network but ip address is provided then port scanner will simply show that the host is down and terminate the program.



Conclusion

The tool can also be expanded to various features like a range of IP or UDP Scanning which will be implemented in future hopefully.