

NETWORK PROGRAMMING

(IS F462)

Project Report

(Group Number : NP 1024)

“ MONITORING TOOL FOR LAB EVALUATION ”

Submitted by

ID NO. :

2016AAPS0245H

2016A7PS0103H

2016AAPS0210H

2015A7PS0028H

NAME:

RANJAN GSK

BHARATH KNS

KAILASH KOLLURU

AJAY DIRASALA



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Birla Institute of Technology and Science Pilani

Hyderabad Campus

Second Semester 2018-19

1. Introduction

1.1 Purpose

The objective of the tool is to monitor the network accesses(intranet and internet) of the clients and send the logs to a hosted server and also give a notification if a client tries to access the local network in real time.The tool also sends a log to the server if a user tries to insert a any device in his respective system and sends a real time notification to the server.

1.2 Abstract

The tool designed takes the server IP address as input to host the server. Each client now uses this IP address to connect to the server and execute the respective scripts.

1.3 What it does?

It simplifies the work of monitoring lab evaluations by recognizing if any USB devices are connected to any of the systems and storing the log of intranet/internet accesses of each system and sends it to the server. This would helping in preventing copying in lab evaluations.

1.4 Overview

The remaining sections of this documentations describes the overall descriptions which includes tool perspective and functions, characteristics of users. It also consists of assumptions and constraints.

2. Brief Description

2.1 User requirements

User is required to have a desktop with Ubuntu and Python installed in it. Along with those the user should have the dependencies and libraries mentioned in the following section installed.

2.2 Assumption and Dependencies

The user needs to install the following libraries and packages for using the tool.

- PsUtil
- Socket
- Os
- Sys
- Time
- IFTop
- Thread
- Subprocess

2.3 Approach

To run this tool we require 3 python scripts namely server.py to host the server,usb.py to detect the usb drives and network.py to monitor the systems networking logs.The server file is hosted on the server systems whereas the remaining two files are hosted on the client to do their respective process.The network.py will require super user permissions to execute without any access errors.

3. Feature description

We initially design a multi client-server system using multithreading concepts. Usage of multi thread is important to access each client's request. Once we set this we use the information from psutil and iftop to send logs from client to server. A local copy is saved on the client.

3.1 Accessing usb drivers and showing real time notification:

The tool uses psutil to detect the media drives. A real time notification is also displayed on the server when a usb driver is inserted by any of the clients. psutil is a cross-platform library for retrieving information on running processes and system utilization (CPU, memory, disks, network, sensors) in Python. It is useful mainly for system monitoring, profiling, limiting process resources and the management of running processes. It implements many functionalities offered by UNIX command line tools such as: ps, top, lsof, netstat, ifconfig, who, df, kill, free, nice, ionice, iostat, iotop, uptime, pidof, tty, taskset, pmap. psutil currently supports the following platforms:

- Linux
- Windows
- macOS
- FreeBSD, OpenBSD, NetBSD
- Sun Solaris
- AIX

We can install the library using `pip install psutil`. For the tool, we used the function `psutil.disk_partitions(all=False)` which returns all mounted disk

partitions as a list of named tuples including device, mount point and filesystem type, similarly to “df” command on UNIX. If all parameter is False it tries to distinguish and return physical devices only (e.g. hard disks, cd-rom drives, USB keys) and ignore all others.

3.1.1 Steps for execution:

- The first step is to start the server(server.py) on them main system with the server IP address and the port number, an image has been attached as a reference to this step in the following section.
- Then the USB script(usb.py) should be executed on the client's system with server IP address and the port number.
- Now when a USB is connected to the client the server gets a notification that a USB has been inserted on 'server IP address'.
- The complete log is sent to the server with the client IP address as the folder name and the time as the name of the log text file which contains the log details.

The names of the files are visible in the screenshots attached below which can be referred to while executing them.

The following screenshots show an overview of we detect the usb drives:

```
gsk98@gsk98-Inspiron-5559: ~/Documents/np_project
File Edit View Search Terminal Help
gsk98@gsk98-Inspiron-5559:~/Documents/np_project$ python server.py 192.168.0.137 8080
```

(Starting the server)

```
es Terminal Tue 06:46
bharath@bharathkns: ~/Downloads
File Edit View Search Terminal Help
bharath@bharathkns:~/Downloads$ python usb.py 192.168.0.137 8080
```

(Client code to start the usb file)

```
gsk98@gsk98-Inspiron-5559: ~/Documents/np_project
File Edit View Search Terminal Help
gsk98@gsk98-Inspiron-5559:~/Documents/np_project$ python server.py 192.168.0.137 8080
192.168.0.133 connected
usb device connected by 192.168.0.133
```

(Real time output at the server once we insert the the usb media on client)

```
device=/dev/sdb1,mountpoint=/media/bharath/Seagate Backup Plus
Drive,fstype=fuseblk,opts=rw,nosuid,nodev,relatime,user_id=0,group_id=0,default_permissions,allow_other,blksize=4096.
```

(The following log is stored on server side under the directory with its name as client ip.)

3.2 Creating network monitoring logs(internet and intranet logs), sending them to server and showing a real time notification if an ip tries to access local network

The tool uses iftop for monitoring the network logs from both internet and intranet accesses. iftop is a free software command-line system monitor tool that produces a frequently updated list of network connections. By default, the connections are ordered by bandwidth usage. The iftop website gives the following description: "iftop does for network usage what top(1) does for CPU usage. It listens to network traffic on a named interface and displays a table of current bandwidth usage by pairs of hosts. Iftop can be installed by using `sudo apt install iftop`.

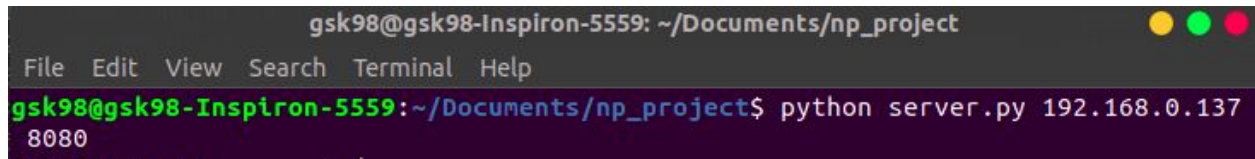
A user needs to be a super user to give all the permissions to run iftop. The tool runs iftop for every 40 seconds and saves the cumulative data usage of the user. These logs are stored in the client and sent to the server at an interval of 40 seconds. By running through these logs server can get if a user has accessed the network. A real time notification appears on the server side if user has accessed any local network connections.

3.2.1 Steps for execution:

- The first step is to start the server(server.py) on the main system with the server IP address and the port number, an image has been attached as a reference to this step in the following section.
- Then the network script(network.py) should be executed on the client's system with server IP address and the port number.
- Now when an intranet access is made by the client the server gets a notification that Network has been accessed by 'client IP address'
- The complete log is sent to the server with the client IP address as the folder name with the log text file which contains the log details of the intranet and internet accesses made by the client.

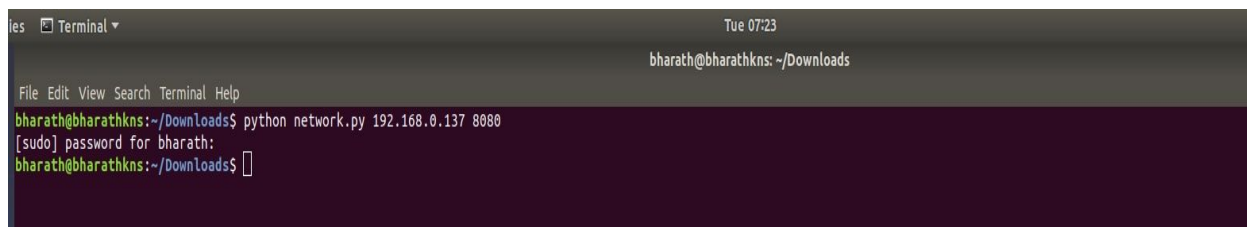
The names of the files are visible in the screenshots attached below which can be referred to while executing them.

The below screenshots show how we implement this and how the logs are being stored:

A terminal window titled 'gsk98@gsk98-Inspiron-5559: ~/Documents/np_project'. The terminal shows the command 'python server.py 192.168.0.137 8080' being entered at the prompt. The window has standard macOS window controls (yellow, green, red buttons) in the top right corner.

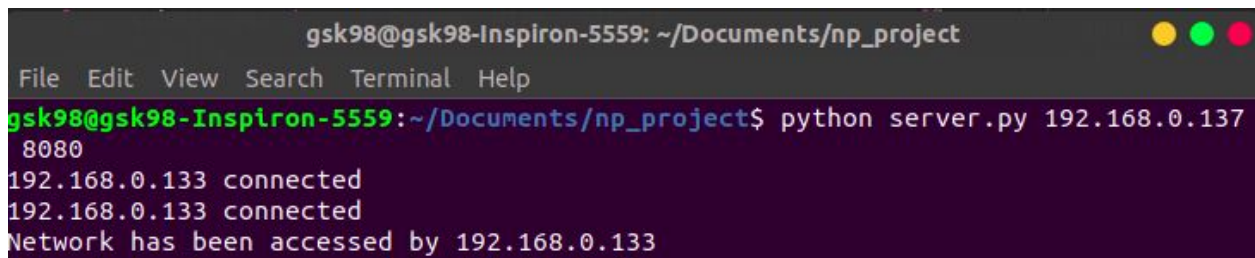
```
gsk98@gsk98-Inspiron-5559: ~/Documents/np_project
File Edit View Search Terminal Help
gsk98@gsk98-Inspiron-5559:~/Documents/np_project$ python server.py 192.168.0.137 8080
```

(Starting the server)

A terminal window titled 'bharath@bharathkns: ~/Downloads'. The terminal shows the command 'python network.py 192.168.0.137 8080' being entered. A password prompt '[sudo] password for bharath:' is visible, followed by a cursor. The window has standard macOS window controls in the top right corner.

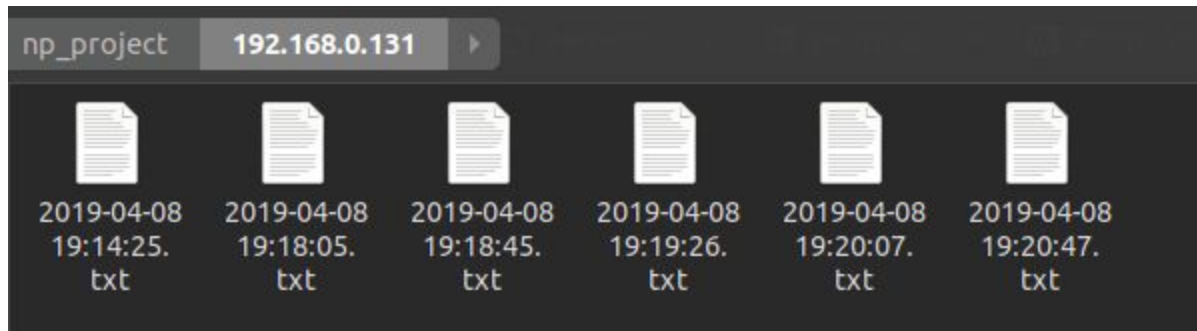
```
bharath@bharathkns:~/Downloads$ python network.py 192.168.0.137 8080
[sudo] password for bharath:
bharath@bharathkns:~/Downloads$
```

(Running the client)

A terminal window titled 'gsk98@gsk98-Inspiron-5559: ~/Documents/np_project'. The terminal shows the command 'python server.py 192.168.0.137 8080' being entered. Below the command, the output shows '192.168.0.133 connected' twice, followed by 'Network has been accessed by 192.168.0.133'. The window has standard macOS window controls in the top right corner.

```
gsk98@gsk98-Inspiron-5559: ~/Documents/np_project
File Edit View Search Terminal Help
gsk98@gsk98-Inspiron-5559:~/Documents/np_project$ python server.py 192.168.0.137 8080
192.168.0.133 connected
192.168.0.133 connected
Network has been accessed by 192.168.0.133
```

(Output on server when client tries to access local network)



(Format of the log files which will be stored in server side with name as their timestamps and directory name as the ip of the client)

```
Interface:
wlp2s0
IP address is:
192.168.0.137
MAC address is: 2c:6e:
85:00:b6:3a
Listening on
wlp2s0
# Host name (port/service if enabled)      last 2s  last 10s  last 40s
cumulative
-----
1 gsk98-Inspiron-5559                      =>      160b    7.63Kb    8.02Kb
32.1KB
maa05s04-in-f14.1e100.net                  <=      500b    1.06Kb    5.68Kb
22.7KB
2 gsk98-Inspiron-5559                      =>         0b    6.45Kb    6.28Kb
25.1KB
maa03s20-in-f14.1e100.net                  <=         0b    831b    1.07Kb
4.26KB
3 gsk98-Inspiron-5559                      =>         0b     96b     30b
120B
sa-in-f188.1e100.net                       <=         0b     0b     0b
0B
4 gsk98-Inspiron-5559                      =>         0b     0b    20b
80B
maa05s06-in-f14.1e100.net                  <=         0b     0b    20b
80B
5 gsk98-Inspiron-5559                      =>         0b     0b    10b
40B
maa05s10-in-f3.1e100.net                   <=         0b     0b    10b
40B
6 gsk98-Inspiron-5559                      =>         0b     0b    10b
40B
maa05s09-in-f3.1e100.net                   <=         0b     0b    10b
40B
```

(The above picture shows an example of log file which is stored in the server. It stores both the internet and the intranet accesses)

4. Conclusion:

The importance of multi threading can be learnt through this exercise as the server needs to address each client separately. Network monitoring tool can be currently used to prevent copying during lab evaluations and this can be further improved to check systems network usage and performance and check for slow or failing systems. This system will save a lot of money and reduce many problems.