**ITP 270 Programming Project 2 Summer 2024**

**(80 Points)**

Programming Project 2 assesses your ability to do the following:

* Create and use Functions
* Use Python’s IO or CSV library to open a datafile for reading and to write to a file.
* Use Python’s list and dictionary data structures to process data.
* Apply algorithms for finding the minimum, maximum, count, and averages for numerical data within a list.

The firewall\_file.txt has 716 rows of data representing packets that were either permitted or denied from entering the destination port.

* The first column represents the source port
* The second column represents the destination port
* The third column represents the action that was taken on the packet either to permit or deny
* The fourth column represents the size of the packet in bytes
* The fifth column represents the number of packets for that firewall entry

We are going to analyze the data to determine the following

* which is the smallest, largest, and average packet size in bytes
* how many and what percentage of the packets are related to ftp data
* how many and what percentage of the packets are related to telnet or secure shell data how many and what percentage of the packets are related to BOOTP or DHCP data
* count how many packets were received for each type of destination port

We will save the statistical information into a dictionary that is then written to a file.

This project is simple to do if we divide it into tasks. The tasks are described in general but do not specify every line of code that you must write. It is your responsibility as a student to analyze the project requirements and write Python code that is needed to meet the requirements.

**Task 1. Create a main() method** that initializes constants for menu choices and initializes an empty list (firewall\_list) that will store data that is read in from the firewall\_file.txt file. Initialize an empty dictionary (firewall\_stats) that will store statistics that are calculated within the processing of the file.

Call a value returning function that will read the firewall\_file.txt data file and save its contents in a two-dimensional list (firewall\_list) for further processing. The firewall\_list should be returned to the main() method. See Task 2 for more information on this function.

Start a loop that will control the running of the program until the user chooses 99 to quit.

Call a function that will display the menu to the user and accept their choice and validate it to make sure that it is either 1, 2, 3, 4, 5, 6, or 99, and then return the valid choice to the main menu. See Task 3 for more information about this function.

Within the main() menu, include a decision structure that will do the following:

If the user enters 1, a function will be called to calculate descriptive statistics for bytes. See Task 4.

If the user enters 2, a function will be called to analyze FTP traffic. See Task 5.

If the user enters 3, a function will be called to analyze Telnet and Secure Shell traffic. See Task 6.

If the user enters 4, a function will be called to analyze DHCP and BOOTP traffic. See Task 7.

If the user enters 5, a function will be called to count packets for each port. See Task 8.

If the user enters 6, a function will be called to save the descriptive statistics to a file. See Task 9.

**Task 2. Create a value returning function that will read the file** into the firewall\_list and return the firewall\_list to the main() method. Print out the number of rows within the firewall\_list to ensure that 716 rows of data have been read in. (You can also print out the list to see if the data is there and then comment out the print line once you are sure that you are reading the file contents into the firewall\_list correctly.)

**Task 3. Create a value returning function that will display the menu** to the user and return a valid menu choice.

The menu choices that the user will see look like this:

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

The user choice must be validated to ensure it is either 1, 2, 3, 4, 5, 6, or 99. An invalid message is displayed to the user to let them know when they’ve entered an invalid menu choice. The menu is then displayed again to let the user see the menu choices. Once a valid choice is entered, it is returned to the main() method.

**Task 4. Create a value returning function that will determine the least bytes, most bytes, and average bytes** of all of the packets within the firewall\_list. You will need a loop to loop through the data and use an algorithm that will help you to determine the smallest byte size, largest byte size, and average byte size of all the packets. You should get the results shown below:

Most Bytes 594

Least Bytes 60

Average Bytes 77.2

Print the results in 2 columns using print formatting. Write the results in key:value pairs to a dictionary firewall\_stats that will be returned to the main method. To see if you are writing to the firewall\_stats dictionary, you can print its contents after writing to the firewall\_stats dictionary, and at least for the data regarding byte sizes, you’d see the following:

{'Most Bytes': 594, 'Least Bytes': 60, 'Average Bytes': 77.16759776536313}

If you see those values written to the dictionary, then you can comment out the print line for the firewall\_stats dictionary before continuing. Remember the firewall\_stats dictionary must be returned to the main() method.

**Task 5. Create a value returning function that will analyze the ftp traffic for ports 20 and 21.** You will need a loop to loop through the data within the firewall\_list. Within the loop, you will need an if statement to filter traffic for ports 20 and 21. Print the rows for the ftp packets as shown below in 6 columns. Notice that the first column is counting the rows within the list.

Row Source Dest. Action Bytes Packets

173 26048 21 deny 60 1

174 26048 21 deny 60 1

175 26048 21 deny 60 1

176 6666 21 deny 60 1

177 43916 21 deny 60 1

178 43916 21 deny 60 1

179 43916 21 deny 60 1

180 26048 21 deny 60 1

181 43916 21 deny 60 1

182 43916 21 deny 60 1

183 43916 21 deny 60 1

184 43916 21 deny 60 1

185 43916 21 deny 60 1

186 43916 21 deny 60 1

187 62908 21 deny 60 1

188 62907 21 deny 60 1

189 6666 21 deny 60 1

190 6666 21 deny 60 1

Count the number of ftp related packets within the if structure as well. After the loop has ended, calculate the Percent of FTP packets by dividing the count of the ftp packets by the length of the firewall\_list and multiply by 100. Print the Count of the FTP Packets and Percent FTP packets as shown below using print formatting in 2 columns:

Count FTP 18

Percent FTP 2.51397

Write the results in key:value pairs to a dictionary firewall\_stats that will be returned to the main method. To see if you are writing to the firewall\_stats dictionary, you can print its contents after writing to the firewall\_stats dictionary, and at least for the data regarding FTP traffic, you’d see the following somewhere within the firewall\_stats dictionary:

'Count FTP': 18, 'Percent FTP': 2.5139664804469275.

If you see those values written to the dictionary, then you can comment out the print line for the firewall\_stats dictionary before continuing. Please note, that you may also see the Most Bytes, Least Bytes, Average Bytes, or other data if you ran that menu choice or another menu choice before trying this menu choice. Remember the firewall\_stats dictionary must be returned to the main() method.

**Task 6. Create a value returning function that will analyze the telnet and secure shell traffic for ports 22 and 23.** You will need a loop to loop through the data within the firewall\_list. Within the loop, you will need an if statement to filter traffic for ports 22 and 23. Print the rows for the telnet and secure shell packets as shown below in 6 columns. Notice that the first column is counting the rows within the list. Rather than showing all of the 427 rows of telnet and secure shell traffic below, I am omitting all of the rows except for the first 5 rows and the last 5 rows. It is okay for you to print all 427 rows.

191 51414 22 deny 60 1

192 36005 22 deny 60 1

193 49220 22 deny 60 1

194 51414 22 deny 60 1

195 51414 22 deny 60 1

…..

613 17916 23 deny 60 1

614 7916 23 deny 60 1

615 25363 23 deny 60 1

616 29022 23 deny 60 1

617 53358 23 deny 60 1

Count the number of Telnet or Secure Shell related packets within the if structure as well. After the loop has ended, calculate the Percent of Telnet or Secure Shell packets by dividing the count of the Telnet or Secure Shell packets by the length of the firewall\_list and multiply by 100. Print the Count of the Telnet or Secure Shell and Percent of the Telnet or Secure Shell packets as shown below using print formatting in 2 columns:

Count Telnet or SSH 427

Percent Telnet or SSH 59.63687

Write the results in key:value pairs to a dictionary firewall\_stats that will be returned to the main method. To see if you are writing to the firewall\_stats dictionary, you can print its contents after writing to the firewall\_stats dictionary, and at least for the data regarding Telnet or Secure Shell traffic, you’d see the following somewhere within the firewall\_stats dictionary:

'Count Telnet or SSH': 427, 'Percent Telnet or SSH': 59.63687150837988

If you see those values written to the dictionary, then you can comment out the print line for the firewall\_stats dictionary before continuing. Please note, that you may also see the Most Bytes, Least Bytes, Average Bytes, or other data if you ran that menu choice or another menu choice before trying this menu choice. Remember the firewall\_stats dictionary must be returned to the main() method.

**Task 7. Create a value returning function that will analyze the BOOTP and DHCP traffic for ports 67 and 68.** You will need a loop to loop through the data within the firewall\_list. Within the loop, you will need an if statement to filter traffic for ports 67 and 68. Print the rows for the BOOTP and DHCP packets as shown below in 6 columns. Notice that the first column is counting the rows within the list.

671 68 67 deny 346 1

672 40394 67 deny 60 1

673 36484 67 deny 60 1

674 68 67 deny 594 1

675 68 67 deny 346 1

676 68 67 deny 594 1

677 68 67 deny 346 1

678 68 67 deny 346 1

679 68 67 deny 346 1

680 56874 67 deny 60 1

681 47827 67 deny 60 1

682 53902 67 deny 60 1

683 40308 67 deny 60 1

684 42223 67 deny 60 1

685 36893 67 deny 60 1

686 68 67 deny 594 1

687 45654 67 deny 60 1

688 55248 67 deny 60 1

689 35644 67 deny 60 1

690 39839 67 deny 60 1

691 68 67 deny 594 1

692 60223 67 deny 60 1

693 67 68 deny 346 1

694 67 68 deny 346 1

-----------------------------------------------------------------

Count the number of BOOTP or DHCP related packets within the if structure as well. After the loop has ended, calculate the Percent of BOOTP or DHCP packets by dividing the count of the BOOTP or DHCP packets by the length of the firewall\_list and multiply by 100. Print the Count of the BOOTP or DHCP and Percent of the BOOTP or DHCP packets as shown below using print formatting in 2 columns:

Count Bootp or DHCP 24

Percent Bootp or DHCP 3.35196

Write the results in key:value pairs to a dictionary firewall\_stats that will be returned to the main method. To see if you are writing to the firewall\_stats dictionary, you can print its contents after writing to the firewall\_stats dictionary, and at least for the data regarding BOOTP and DHCP traffic, you’d see the following somewhere within the firewall\_stats dictionary:

'Count Bootp or DHCP': 24, 'Percent Bootp or DHCP': 3.35195530726257

If you see those values written to the dictionary, then you can comment out the print line for the firewall\_stats dictionary before continuing. Please note, that you may also see the Most Bytes, Least Bytes, Average Bytes, or other data if you ran that menu choice or another menu choice before trying this menu choice. Remember the firewall\_stats dictionary must be returned to the main() method.

**Task 8. Count the number of rows received for each destination port type.**

There are several different ways you can approach this. Here’s what I did to practice with sets and dictionaries.

I created an empty set. I used a for loop to increment through the firewall\_list at each row for column 1 which is where the destination ports are at. Within the for loop, I created a variable called item which saves the value of the firewall\_list for the row that is being iterated and column 1. I used the set method add() to add the item to the set. Because sets don’t allow duplicate items, you will get a set that has the unique port numbers represented within the firewall\_file.

After exiting the for loop, I created a dictionary that uses the dict.fromkeys() method passing the name of the set to be used as the dictionary keys, and the value of 0 to be set for each value for each dictionary key. (We’ll change those 0s to the actual count of the number of rows for each port type next.). Remember that dictionaries don’t allow duplicate keys either.

Then, I set up a for loop that loops for each key in the dictionary. Within it, I set a counter = 0 that is counting the number of rows for the key. Then I started an inner for loop (meaning it is nested) inside the outer for loop that is looping for each row in the firewall. Within the inner for loop, I used an if statement to compare the firewall\_list row and then column 1 to see if it is equal to the key from the outer loop. If those values are equal, then I incremented the counter. Then I set the value for each key equal to the counter. After the outer for loop ends, I set up a for loop that then loops through the dictionary showing each key (which is a port number) and the count of rows for it which is the counter that was saved for that key. Here are what the results look like:

Port # Traffic Count

0 88

1 65

67 22

68 2

37 12

70 1

80 7

17 19

81 13

21 18

22 81

23 346

53 29

25 12

88 1

(As a sanity check, I summed up the values for the dictionary and they totaled 716 which matches the number of rows, and then commented out the print statement afterwards.)

**Task 9. Write the firewall\_stats dictionary to a firewall\_stats.csv file.**

I used Python’s CSV library to write the contents of the firewall\_stats dictionary to a firewall\_stats.csv file. When I open the firewall\_stats.csv file in a text editor or IDE, I see the following results. Depending on how you test your code, you may see the results appear in a different order which is fine. The point is that you have the stats from Tasks 4-7 written within the firewall\_stats dictionary which are then written to the firewall\_stats.csv file.

Most Bytes,594

Least Bytes,60

Average Bytes,77.16759776536313

Count FTP,18

Percent FTP,2.5139664804469275

Count Telnet or SSH,427

Percent Telnet or SSH,59.63687150837988

Count Bootp or DHCP,24

Percent Bootp or DHCP,3.35195530726257

To better your understanding of the project requirements, here is the **How The Program Should Run** section that shows test data in ***Red, Bolded, and Italicized***.

This system performs analysis on a datafile containing firewall data.

There are 716 rows of data in the firewall list.

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***0***

Invalid Choice.

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***7***

Invalid Choice.

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***1***

Statistics on Byte Sizes

Most Bytes 594

Least Bytes 60

Average Bytes 77.2

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***2***

Destination Ports 20 and 21 (FTP)

Row Source Dest. Action Bytes Packets

173 26048 21 deny 60 1

174 26048 21 deny 60 1

175 26048 21 deny 60 1

176 6666 21 deny 60 1

177 43916 21 deny 60 1

178 43916 21 deny 60 1

179 43916 21 deny 60 1

180 26048 21 deny 60 1

181 43916 21 deny 60 1

182 43916 21 deny 60 1

183 43916 21 deny 60 1

184 43916 21 deny 60 1

185 43916 21 deny 60 1

186 43916 21 deny 60 1

187 62908 21 deny 60 1

188 62907 21 deny 60 1

189 6666 21 deny 60 1

190 6666 21 deny 60 1

Count FTP 18

Percent FTP 2.51397

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***3***

Destination Ports 22 and 23 (Secure Shell and Telnet)

191 51414 22 deny 60 1

192 36005 22 deny 60 1

193 49220 22 deny 60 1

194 51414 22 deny 60 1

195 51414 22 deny 60 1

So that you don’t have to read page after page of Secure Shell and Telnet rows, I omitted showing all rows except for the first 5 rows and last 5 rows in these instructions. It is okay for you to print all 427 rows in your submission.

613 17916 23 deny 60 1

614 7916 23 deny 60 1

615 25363 23 deny 60 1

616 29022 23 deny 60 1

617 53358 23 deny 60 1

Count Telnet or SSH 427

Percent Telnet or SSH 59.63687

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***4***

Destination Ports 67 and 68 (BOOTP and DHCP)

671 68 67 deny 346 1

672 40394 67 deny 60 1

673 36484 67 deny 60 1

674 68 67 deny 594 1

675 68 67 deny 346 1

676 68 67 deny 594 1

677 68 67 deny 346 1

678 68 67 deny 346 1

679 68 67 deny 346 1

680 56874 67 deny 60 1

681 47827 67 deny 60 1

682 53902 67 deny 60 1

683 40308 67 deny 60 1

684 42223 67 deny 60 1

685 36893 67 deny 60 1

686 68 67 deny 594 1

687 45654 67 deny 60 1

688 55248 67 deny 60 1

689 35644 67 deny 60 1

690 39839 67 deny 60 1

691 68 67 deny 594 1

692 60223 67 deny 60 1

693 67 68 deny 346 1

694 67 68 deny 346 1

Count Bootp or DHCP 24

Percent Bootp or DHCP 3.35196

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***5***

Traffic Counts for each Destination Port

Port # Traffic Count

0 88

1 65

67 22

68 2

37 12

70 1

80 7

17 19

81 13

21 18

22 81

23 346

53 29

25 12

88 1

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***6***

The fw\_stats\_file.csv has been updated.

-----------------------------------------------------------------

Please choose from the following menu:

Enter 1 to calculate descriptive statistics for bytes.

Enter 2 to analyze FTP Traffic.

Enter 3 to analyze Telnet and Secure Shell Traffic.

Enter 4 to analyze DHCP and BOOTP Traffic.

Enter 5 to count packets for each port.

Enter 6 to save the descriptive statistics to a file.

Enter 99 to Quit.

-----------------------------------------------------------------

***99***

**Please submit the firewall\_file.py script for Programming Project 2. Refer to Programming Labs 3, 4, and 5 and the Extra Credit for Programming Project 2 for code that is useful for coding this project.**