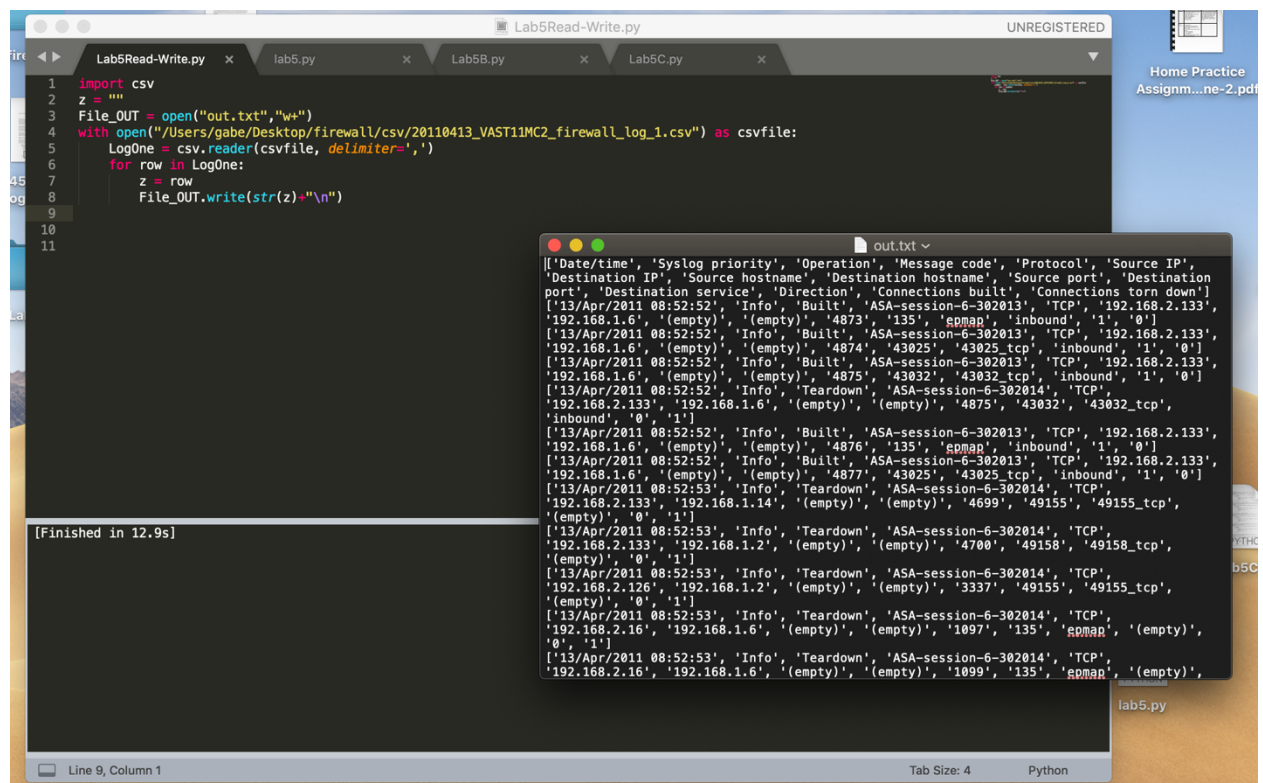


Option 2 – Programming

Part A:

Below is a screenshot of my program going through the CSV file for the first log and reproducing the information, line for line, in the output.txt file. This basic program could easily run through every single log file and reproduce them line by line with some slight modification; however, I chose to only do it for the first log file since the file for all 5 logs would be massive and I would be unable to open it. I could also output each log into its own unique out.txt file following the same logic below.



The screenshot shows a Python script named `Lab5Read-Write.py` running in a terminal window. The script is processing a CSV file named `20110413_VAST11MC2_firewall_log_1.csv` and outputting the data to a file named `out.txt`. The output file contains a list of log entries, each represented as a list of strings. The log entries include timestamps, syslog priorities, operations, message codes, protocols, source and destination IP addresses, source and destination ports, destination services, directions, connections built, and connections torn down. The script is currently at line 9, column 1, and has finished in 12.9 seconds.

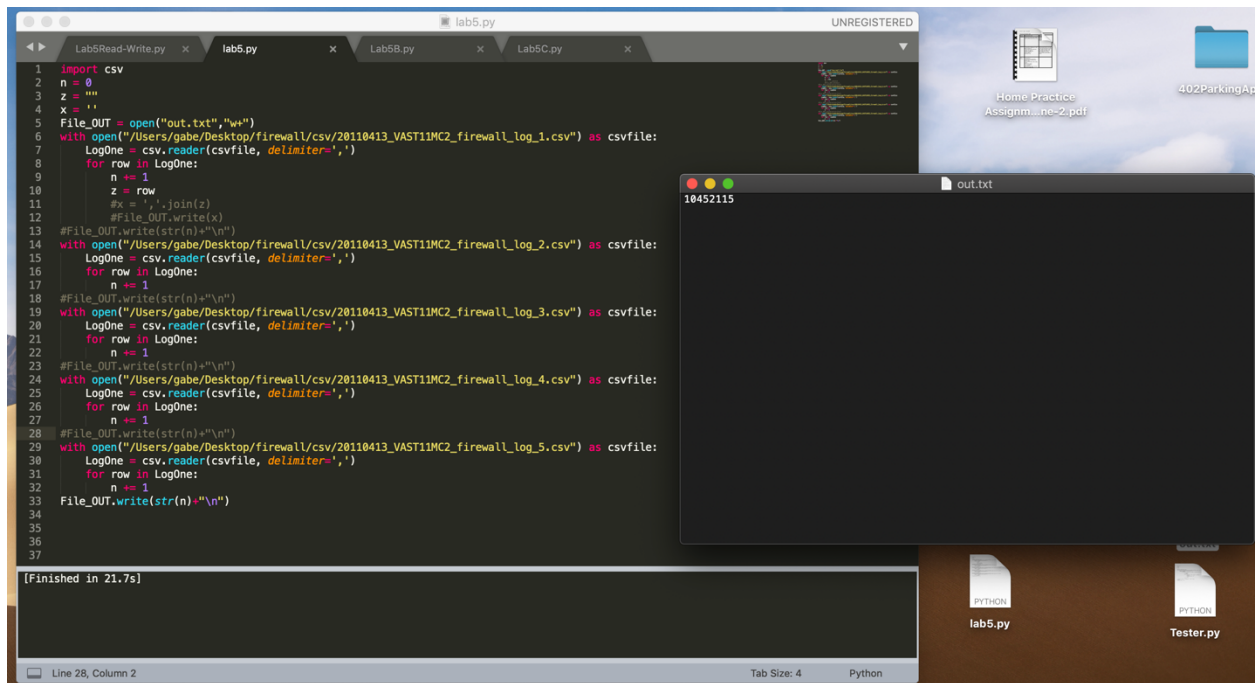
```
1 import csv
2 z = ""
3 File_OUT = open("out.txt", "w")
4 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_1.csv") as csvfile:
5     LogOne = csv.reader(csvfile, delimiter=',')
6     for row in LogOne:
7         z = row
8         File_OUT.write(str(z)+"\n")
9
10
11
```

[Finished in 12.9s]

out.txt

```
[['Date/time', 'Syslog priority', 'Operation', 'Message code', 'Protocol', 'Source IP',
'Destination IP', 'Source hostname', 'Destination hostname', 'Source port', 'Destination
port', 'Destination service', 'Direction', 'Connections built', 'Connections torn down']
['13/Apr/2011 08:52:52', 'Info', 'Built', 'ASA-session-6-302013', 'TCP', '192.168.2.133',
'192.168.1.6', '(empty)', '(empty)', '4873', '135', 'epmap', 'inbound', '1', '0']
['13/Apr/2011 08:52:52', 'Info', 'Built', 'ASA-session-6-302013', 'TCP', '192.168.2.133',
'192.168.1.6', '(empty)', '(empty)', '4874', '43025', '43025_tcp', 'inbound', '1', '0']
['13/Apr/2011 08:52:52', 'Info', 'Built', 'ASA-session-6-302013', 'TCP', '192.168.2.133',
'192.168.1.6', '(empty)', '(empty)', '4875', '43032', '43032_tcp', 'inbound', '1', '0']
['13/Apr/2011 08:52:52', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.133', '192.168.1.6', '(empty)', '(empty)', '4875', '43032', '43032_tcp',
'inbound', '0', '1']
['13/Apr/2011 08:52:52', 'Info', 'Built', 'ASA-session-6-302013', 'TCP', '192.168.2.133',
'192.168.1.6', '(empty)', '(empty)', '4876', '135', 'epmap', 'inbound', '1', '0']
['13/Apr/2011 08:52:52', 'Info', 'Built', 'ASA-session-6-302013', 'TCP', '192.168.2.133',
'192.168.1.6', '(empty)', '(empty)', '4877', '43025', '43025_tcp', 'inbound', '1', '0']
['13/Apr/2011 08:52:53', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.133', '192.168.1.14', '(empty)', '(empty)', '4699', '49155', '49155_tcp',
'(empty)', '0', '1']
['13/Apr/2011 08:52:53', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.133', '192.168.1.2', '(empty)', '(empty)', '4700', '49158', '49158_tcp',
'(empty)', '0', '1']
['13/Apr/2011 08:52:53', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.126', '192.168.1.2', '(empty)', '(empty)', '3337', '49155', '49155_tcp',
'(empty)', '0', '1']
['13/Apr/2011 08:52:53', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.16', '192.168.1.6', '(empty)', '(empty)', '1097', '135', 'epmap', '(empty)',
'0', '1']
['13/Apr/2011 08:52:53', 'Info', 'Teardown', 'ASA-session-6-302014', 'TCP',
'192.168.2.16', '192.168.1.6', '(empty)', '(empty)', '1099', '135', 'epmap', '(empty)',
```

The below code goes through each file and tallies the total number of connections. In the output file it prints the total number of 10,452,115 connections. I could also vary this program to print the total amount of connections contained within each log file individually.



The screenshot shows a Python IDE with a script named `lab5.py` and an output window. The script reads five CSV files from a directory and tallies the total number of connections. The output window displays the result `10452115`.

```
1 import csv
2 n = 0
3 z = ""
4 x = ""
5 File_OUT = open("out.txt", "w")
6 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_1.csv") as csvfile:
7     LogOne = csv.reader(csvfile, delimiter=',')
8     for row in LogOne:
9         n += 1
10        z = row
11        #x = ','.join(z)
12        #File_OUT.write(x)
13    #File_OUT.write(str(n)+"\n")
14    with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_2.csv") as csvfile:
15        LogOne = csv.reader(csvfile, delimiter=',')
16        for row in LogOne:
17            n += 1
18        #File_OUT.write(str(n)+"\n")
19        with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_3.csv") as csvfile:
20            LogOne = csv.reader(csvfile, delimiter=',')
21            for row in LogOne:
22                n += 1
23            #File_OUT.write(str(n)+"\n")
24            with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_4.csv") as csvfile:
25                LogOne = csv.reader(csvfile, delimiter=',')
26                for row in LogOne:
27                    n += 1
28                #File_OUT.write(str(n)+"\n")
29                with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_5.csv") as csvfile:
30                    LogOne = csv.reader(csvfile, delimiter=',')
31                    for row in LogOne:
32                        n += 1
33                File_OUT.write(str(n)+"\n")
34
35
36
37
```

[Finished in 21.7s]

Line 28, Column 2

Tab Size: 4

Python

10452115

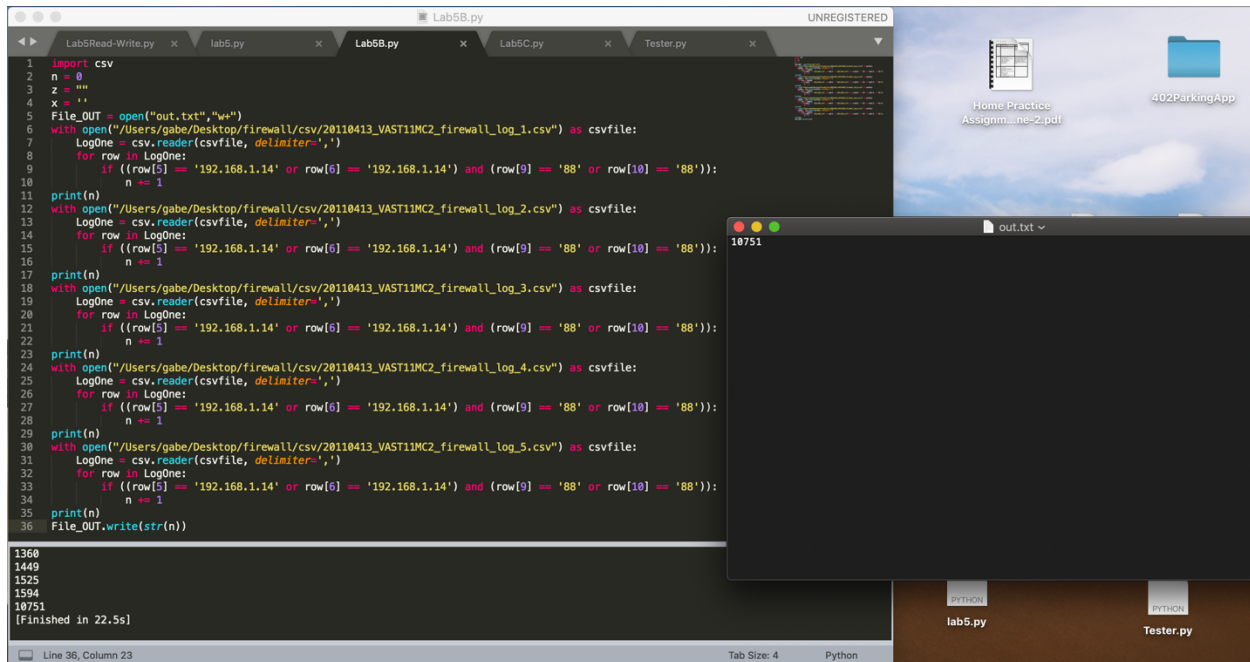
out.txt

lab5.py

Tester.py

Part B:

Below is the code that goes through each document and counts the total number of connections that fit the specified criteria (containing IP 192.168.1.14 and port number 88 with both destination and source ip and destination and source port). The code then creates a running total of the amount of connections that fit this criteria and outputs only the total amount of connections to the out.txt file. Within the program itself it keeps a running count as it goes through each document which I used for testing purposes. Furthermore, I could add an additional line in each for loop iteration that prints out the whole row of the information desired.



```
1 import csv
2 n = 0
3 z = ""
4 x = ""
5 File_OUT = open("out.txt", "w")
6 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_1.csv") as csvfile:
7     LogOne = csv.reader(csvfile, delimiter=',')
8     for row in LogOne:
9         if ((row[5] == '192.168.1.14' or row[6] == '192.168.1.14') and (row[9] == '88' or row[10] == '88')):
10             n += 1
11 print(n)
12 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_2.csv") as csvfile:
13     LogOne = csv.reader(csvfile, delimiter=',')
14     for row in LogOne:
15         if ((row[5] == '192.168.1.14' or row[6] == '192.168.1.14') and (row[9] == '88' or row[10] == '88')):
16             n += 1
17 print(n)
18 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_3.csv") as csvfile:
19     LogOne = csv.reader(csvfile, delimiter=',')
20     for row in LogOne:
21         if ((row[5] == '192.168.1.14' or row[6] == '192.168.1.14') and (row[9] == '88' or row[10] == '88')):
22             n += 1
23 print(n)
24 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_4.csv") as csvfile:
25     LogOne = csv.reader(csvfile, delimiter=',')
26     for row in LogOne:
27         if ((row[5] == '192.168.1.14' or row[6] == '192.168.1.14') and (row[9] == '88' or row[10] == '88')):
28             n += 1
29 print(n)
30 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_5.csv") as csvfile:
31     LogOne = csv.reader(csvfile, delimiter=',')
32     for row in LogOne:
33         if ((row[5] == '192.168.1.14' or row[6] == '192.168.1.14') and (row[9] == '88' or row[10] == '88')):
34             n += 1
35 print(n)
36 File_OUT.write(str(n))
37
1360
1449
1525
1594
10751
[Finished in 22.5s]
```

out.txt

10751

Line 36, Column 23 Tab Size: 4 Python

lab5.py Tester.py

Part C:

Below is a screencap of the program running and creating the table of the amount of times that IP 172.20.1.5 and port 80(for both source and destination IP as well as source and destination port) appear and how many times per hour. This program goes through and creates a count of the amount of times the criteria appears while constantly checking that the hour matches. Once the code sees that the hour no longer matches, or has changed, it outputs the total for the hour it was keeping track of and resets the counter back to zero. Then it goes through and starts counting the amount of times the criteria match for the new hour. It is important to note that the first line in my output is 0 because of the way I created my buffer variable for recording and iterating through different hours.

```
1 import csv
2 n = 0
3 q = ''
4 w = ''
5 z = ''
6 x = ''
7 File_OUT = open("out.txt", "w")
8 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_1.csv") as csvfile:
9     LogOne = csv.reader(csvfile, delimiter=',')
10     for row in LogOne:
11         if ((row[5] == '172.20.1.5' or row[6] == '172.20.1.5') and (row[9] == '80' or row[10] == '80')):
12             z = row[0]
13             x = z[12:14]
14             if x != q:
15                 w = z[0:11] + "T0" + str(q) + ' 172.20.1.5 ' + '80 ' + str(n)
16                 print(w)
17                 File_OUT.write(w + '\n')
18                 n = 0
19                 q = x
20             n += 1
21 #w = z[0:11] + "T0" + q + ' 172.20.1.5 ' + '80 ' + str(n)
22 #print(w)
23 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_2.csv") as csvfile:
24     LogOne = csv.reader(csvfile, delimiter=',')
25     for row in LogOne:
26         if ((row[5] == '172.20.1.5' or row[6] == '172.20.1.5') and (row[9] == '80' or row[10] == '80')):
27             z = row[0]
28             x = z[12:14]
29             if x != q:
30                 w = z[0:11] + "T0" + str(q) + ' 172.20.1.5 ' + '80 ' + str(n)
31                 print(w)
32                 File_OUT.write(w + '\n')
33                 n = 0
34                 q = x
35             n += 1
36 #w = z[0:11] + "T0" + q + ' 172.20.1.5 ' + '80 ' + str(n)
37 with open("/Users/gabe/Desktop/firewall/csv/20110413_VAST11MC2_firewall_log_3.csv") as csvfile:
38     LogOne = csv.reader(csvfile, delimiter=',')
39     for row in LogOne:
40         if ((row[5] == '172.20.1.5' or row[6] == '172.20.1.5') and (row[9] == '80' or row[10] == '80')):
41             z = row[0]
42             x = z[12:14]
43             if x != q:
44                 w = z[0:11] + "T0" + str(q) + ' 172.20.1.5 ' + '80 ' + str(n)
45                 print(w)
46                 File_OUT.write(w + '\n')
47                 n = 0
48                 q = x
49             n += 1
50 #w = z[0:11] + "T0" + q + ' 172.20.1.5 ' + '80 ' + str(n)
51 #print(w)
52 File_OUT.close()
53 print("Finished in 27.6s")
```

13/Apr/2011T0 172.20.1.5 80 0
13/Apr/2011T011 172.20.1.5 80 2896391
13/Apr/2011T012 172.20.1.5 80 7898854