Project: Profiling Internet User

Course: CIS6930.009S19 Information Security and Privacy in Dist. System
Instructor: Dr. Sriram Chellappan

Submitted By: Fariha Moomtaheen
UID: 3053 6417

Introduction

This project is about profiling multiple subjects based on their internet usages and making statistical distinction among them. 3 time windows have been chosen to observe the profiling criteria- 10 second, 227 seconds, and 5 minutes.

Data:

The source data is derived from the internet usage history of 54 users over 1 month period from a IP traffic collecting software named CISCO NetFlow(5). Each user has a unique file and the columns are described as below-

unix_secs-	Current count of seconds since 0000 UTC 1970
sysuptime	Current time in milliseconds since the export device booted
dpkts	Packets in the flow
doctets	Total number of Layer 3 bytes in the packets of the flow
doctets/dpkts	**Profiling Criteria**
Real First Packet	Initial date and time of each flow in epoch format
Real End Packet	End date and time of each flow in epoch format
first	SysUptime at start of flow
last	SysUptime at the time the last packet of the flow was received
Duration	last - first

(Definitions from- https://www.cisco.com/c/en/us/td/docs/net_mgmt/netflow_collection_engine/3-6/user/guide/format.html)

Profiling Process:

The process can be divided into two parts- writing code to generate usable values and analysing those values to come to a decision.

Data Splitting: The requirement of the project was to take only 1st two week's data and split them into windows. We take only monday to friday data from 8.00 am to 5.00 pm. But not all user's data start from monday. In order to make sure I am starting from monday, I take the 1st Real First Packet value and calculate how far behind that day is from Monday. Then adjust the days to define start time from the upcoming monday at 8.00 am.

By adding 7 days to the 1st Monday, the start time of the 2nd week can be obtained. After getting the start times of these two weeks, I call a function called *create_weeksheet*() twice to make two separate dataframes for each week, per user, using the required window time. The dataframe looks like this-

Index	Weekday	fromTime	toTime	Time	Octets/Duration
0	Monday	2013-02-04 08:00:00	2013-02-04 08:00:10	08:00:00AM-08:00:10AM	0
1	Monday	2013-02-04 08:00:10	2013-02-04 08:00:20	08:00:10AM-08:00:20AM	0
2	Monday	2013-02-04 08:00:20	2013-02-04 08:00:30	08:00:20AM-08:00:30AM	0
3	Monday	2013-02-04 08:00:30	2013-02-04 08:00:40	08:00:30AM-08:00:40AM	0
4	Monday	2013-02-04 08:00:40	2013-02-04 08:00:50	08:00:40AM-08:00:50AM	0
5	Monday	2013-02-04 08:00:50	2013-02-04 08:01:00	08:00:50AM-08:01:00AM	0
6	Monday	2013-02-04 08:01:00	2013-02-04 08:01:10	08:01:00AM-08:01:10AM	0
7	Monday	2013-02-04 08:01:10	2013-02-04 08:01:20	08:01:10AM-08:01:20AM	0

Figure: Dataframe for 10 sec window

Data populating: After I have two dataframes with the specified window divisions, I still have one column left to calculate the doctets/duration values. I make a function called *populate_weeksheet*() for this purpose. It iterates through the Real First Packet values of a user and checks if it falls within the 1st two weeks and whether it has a duration > 0. Then it finds out in which window frame that particular network flow starts flowing by calculating the indexes.

If a window has more than one flows, I take the average for all the doctets/duration values within that window.

After finding doctets/duration for all the users, I save the data in files for further calculation.

Spearman Correlation: In order to calculate correlation among different users and among same user's data over time, spearman correlation coefficient is used. I used python's scipy library in order to calculate the values r1a2a, r1a2b, and r2a2b

Z-value and P-value: I write two separate functions *calculate_zValue*() and *calculate_pValue*(), to find the final P values. Each P value is between one user's 1st and 2nd weeks data vs another user's 1st and 2nd week's data. Except for the diagonals, where the P value is between the same user's data for weeks 1 and 2.

Observation: After implementing the above parts, we have 3 final tables with the P values for 3 window times. There are couple of things that jumps out from observing the tables-

a. The diagonal values for all 3 tables are 0.499 (<= 0.5). Which means, the correlation coefficient is significantly smaller, making them distinct. But this is not expected, as the data are from the same users. This makes the internet usage of the same user across multiple weeks distinct.
b. There are some users for whom all the P values are zeros across both column and row. This is in fact because there there were no data within either the 1st or 2nd week for that particular user. For example, user 5 (file- cjdk88.xlxs) has no internet usage during the 2nd week. So the values couldn't be calculated across weeks against other users.
c. There are some cases where the P values are 1. It is because the correlation among those two subjects has come up as the most, marking them as the same. But none of those 1 values fall in the diagonals, which means this process is profiling separate users as significantly indistinguishable.

Week1/Week2	User1	User2	User3	User4	User5	User6	User7
User1	0.4999999995	0.9151158402	0.9593877589	0.7079628464	0	0.8627336886	0.7092903988
User2	0.55339229	0.4999999995	0.5420160631	0.3991498531	0	0.4913116426	0.4967789343
User3	0.8089733986	0.921821651	0.4999999995	0.8138892788	0	0.3530395932	0.6566498206
User4	0.4218463059	0.3899442202	0.3099522339	0.4999999995	0	0.1545257809	0.5591069791
User5	0	0	0	0	0	0	
User6	0.2007093301	0.5247890146	0.4406799021	0.4779443463	0	0.4999999995	0.2088651408
User7	0.9694814682	0.9796688298	0.971656355	0.9543521963	0	0.8462684698	0.4999999995

Figure: Sample data for 7 users for time window = 5 minutes.

Analysis: I write another code to find the average number of matches for three time windows. Matches indicate the values that are > 0.5 among the 54*54 = 2916 values. It also calculates the individual number of matches for each time window and the percentage of matches.

10 sec window: match 1773 (60.8%)
227 sec window: match 1571 (53.9%)
5 min window: match 1571 (53.9%)
Average no of matches: 1638.33 (56.2%)

We can see from here, the number of matches for the last 2 windows are same. As we increase the time window, the number of matches decreases, which can mean that it gets better at distinguishing users.

Conclusion:

After implementing all the necessary steps from the project, I can see that bigger window is better in terms of authentication. However, it still doesn't show that, a) each user's data in one week is statistically indistinguishable from the same user's data across another week, and b) the number of other users whose data is statistically indistinguishable from a particular user is minimum (ideally 0). We are still getting a 56% match for the 5 min window. This is because 1 month is not sufficient to correctly observe and predict a behavior. Maybe, with more data we could properly show the trend and decide the optimal time window.