Name: Jordan Cheong

Subject: CSCI262

Assignment: 3

The brief is to design and implement an email IDS and event modeller. The language I have chosen is python.

Packages required: numpy, installed with “pip install numpy”

**Initial input**

The format of the initial input is given in the command line.

The format is python3 (filename) (text file1) (text file2) (number of days). An example is

“python3 asn3.py Events.txt Stats.txt 8”

The files should be in a specific format.

The information in the files are stored as multi-dimensional arrays, as they provide convenient ways of storing data while being adaptable to large amounts of data.

There are 2 potential inconsistencies identified, both of which are checked in the method “checkInconsistencies”, on lines 91-109

|  |  |
| --- | --- |
| Number | Inconsistency |
| 1 | Value given in the first line, which identifies number of lines(events) in the files, differ. |
| 2 | Event names differ. |

**Activity Engine**

The activity engine takes the numbers provided in the two text files provided to generate events within the given bounds. The specific distribution used is gaussian distribution as mean and standard deviation is provided. There is no distinction made between discrete and continuous events, other than the rounding used. Continuous events are rounded to 2 decimal places, while discrete events are integers.

In addition, the files from this initial activity engine are logged, which will be stored as “eventlog.txt”, in the format

Day (Number)

Event1: (Number)

Event2 : (Number)

And so on.

**Analysis Engine**

The program then proceeds to analyse the generated data, calculating the mean and standard deviation, and storing those values in analysisLog.txt

**Alert Engine**

Then, the program requests for a text file and a number of days from the user.

This new file should be formatted similarly to the second file in the initial command line run of the program, and it provides a different set of data (and parameters) to generate events with.

It then compares the newly generated events with the data calculated in the Analysis Engine, and anomaly counts are tallied. According to the weight provided in the first text file, if the threshold is exceeded (2\*sum of weights), an intrusion will be detected, and the user will be notified.

Finally, the user is returned to the start of this phase so another query can be run.