IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 26, Issue 6, Ser. 1 (Nov. – Dec. 2024), PP 08-13
www.iosrjournals.org

Host Based-Intrusion Detection System (Anomaly Detection System)

Omkar Dhananjay Sagade
Dept. of computer engineering
MIT Polytechnic (MSBTE)
Pune, India

Atharva Dnyaneshwar Ubhe
Dept. of computer engineering
MIT Polytechnic (MSBTE)
Pune, India

Prof. Prerna Siddharth Patil
Dept. of computer engineering
MIT Polytechnic (MSBTE)
Pune, India

Abstract
The objective of our statistical anomaly detection project is to develop an IDS that notices behavior which is based on observed action, detects abnormalities or possible intrusions. The system utilizes statistical analysis techniques, specifically features to discover anomalous patterns of activity, and receives input data in the form of stats.txt and event.txt files. The project's goal is to offer a dependable and efficient IDS that can assist in identifying and averting cyberattacks on networks and computer systems.

Keywords—*anomaly; network intrusion; statistical analysis; intrusion detection system; and.txt.*

Date of Submission: 24-10-2024 Date of Acceptance: 04-11-2024

I. Introduction

The creation of a statistical anomaly detection system is the main goal of our research. The system's purpose is to watch how a system or application behaves and identify any irregularities or possible intrusions based on the activity that is seen. In the world of cybersecurity, anomaly detection is an essential procedure that is getting more and more crucial as the quantity of cyberattacks rises. Our project's objective is to offer a dependable and effective method for identifying unusual activity in a system or application. Two input files will be integrated by the system: one will contain statistical information on the program or system, and the other will contain an event log. When the system notices unusual activity, it will send out notifications so that security staff can look into it and take the necessary action.

The "Anomaly based Intrusion Detection System" shows promise since, in light of the constantly changing landscape of cyber threats, there is an increasing demand for strong and efficient network security measures. Technological developments in artificial intelligence and machine learning are anticipated to augment the precision and efficacy of anomaly detection systems, hence facilitating proactive identification. Hence, this is Host based IDS based implemented using theme Computational Intelligence.

Motivation:

The goal of our system project is to develop a system that can efficiently monitor an application's or system's behavior and determine any anomalies or possible intrusions based on the activity that is observed. The growing frequency of cyberattacks and security lapses in the digital realm has made it essential to create sophisticated ids in order to safeguard confidential information and guarantee system stability. Our study uses machine learning algorithms and statistical analysis to deliver an effective and dependable solution to these problems. Our solution can assist in preventing data loss, system outages, and other undesirable effects of security breaches by identifying such threats early.

DOI: 10.9790/0661-2606010813

Objectives:
We will be able to:

1. Create and demonstrate by putting into practice an IDS system that can track an application's or system's behavior in real time.
2. Creating algorithms to examine system or application data and find irregularities or possible intrusions.
3. Including the stats.txt and event.txt files in the IDS system so that it can accept them as input.
4. To notify system administrators in real time of any potential threats or incursions as they are discovered.
5. Assessing the IDS system's performance and looking at its scalability, accuracy, and speed.

Scope:

Our project's goal is to create an intrusion detection system (IDS) that can monitor a system's or application's behavior and identify anomalies or possible intrusions based on the activity that is seen.

The project's goal is to offer a dependable and effective method for identifying security risks and shielding computer systems from malevolent assaults. The scope comprises creating and executing the system architecture, creating the required tools and methods for anomaly detection, and testing and analyzing the system to determine its overall performance. A user-friendly interface for configuring the system and seeing the abnormalities that have been found is also part of the project. The project's scope can be expanded to include more sophisticated threat detection methods and integration with other security systems.

II. Literature Survey

Over the years, a lot of research has been done in the topic of intrusion detection, and several methods and strategies for identifying and stopping harmful activity in computer systems have been presented. Conventional signature-based detection techniques are ineffective against new threats because they depend on patterns of malicious activity that are already known. The development of statistical anomaly-based detection techniques, which are able to recognize departures from expected behavior and uncover as-yet-undiscovered attacks, has so attracted increased attention.[3]

Using machine learning methods for statistical anomaly identification is one well-liked strategy. These techniques usually entail using a sizable dataset of typical behavior to train a model, which is then used to detect departures from the norm in real time. [2] For this, a variety of machine learning methods have been applied with differing degrees of success, including neural networks, support vector machines, and k-nearest neighbors.[1]

Additionally, there has been an increase in interest in using deep learning methods for intrusion detection in recent years. In identifying irregularities in network traffic and system logs, deep learning models like convolutional neural networks and recurrent neural networks have demonstrated encouraging results. [4]

But creating statistical anomaly detection systems that work well requires striking a balance between minimizing false positives and detecting real anomalies.[3] Setting suitable criteria for anomaly detection and fine-tuning the detection algorithms are necessary for this.

Overall, the literature survey highlights the potential of statistical anomaly detection for intrusion detection and the ongoing research in developing more effective and efficient methods for detecting previously unknown attacks.[1]

III. Project Background

Statistical anomaly intrusion detection is the focus of the study. One kind of intrusion detection system that operates by seeing patterns in network traffic that deviate from the norm is anomaly intrusion detection. Numerous attack types, such as port scanning, denial of service assaults, and other forms of intrusion attempts, can be identified by the system.

The project's objective is to create a system that can track an application's or system's behavior and, using the activity it observes, identify abnormalities or possible intrusions. The system's architecture will enable it to examine network traffic and spot unusual trends. By doing this, the system will be able to identify such attacks before they have a chance to do any harm.

The project will examine network traffic data using statistical analysis techniques in order to accomplish this goal. Using a set of typical network traffic patterns as training data, the system will be able to recognize patterns that deviate from the norm. System administrators can use the alert that the system generates when an anomaly is found to be informed of a possible intrusion attempt.

All things considered; the project might offer network security experts a strong tool that could be utilized to help stop a range of different kinds of attacks.

IV. Implementation

Our system appears to be an intrusion detection system. By replicating various events and behaviors, it captures data about them. Using this data to identify anomalies or suspect behavior is a popular use case for intrusion detection systems.The Event Java class is defined in the code. The Event class contains the following private instance variables (or fields): weight, min, max, event Name, is Discrete, and so on. The methods to String(), getMin(), setMin(), getMax(), setMax(), getMinimum(), setMinimum(), getMaximum(), setMaximum(), getWeight(), setWeight(), and isDiscrete() are among the other methods that the Event class has. The toString() method converts the properties of the Event object into a formatted string. The corresponding string representation of the object is returned once the is Discrete attribute has been determined to be true or false. It returns a string containing the event name, type, minimum and maximum values, and weight if is Discrete is true. It returns a string containing the event name, type, minimum and maximum values (as doubles), and weight if is Discrete is false. The Event class has two constructors available. The event name, is discrete, minimum, maximum, and weight attributes are initialized by the first constructor. The event name, is Discrete, min, max, and weight attributes are initialized by the second constructor.

This replicates server activity data and applies statistical analysis to the generated data. A number of techniques are included in the application to carry out various tasks, such as creating simulated activity data, computing statistical measures like mean and standard deviation, and presenting the findings. In order to verify that the input data is consistent, the program's main method calls the read Check Display Files method, which receives data from input files, does some preliminary tests, and then calls the check File Inconsistency function. The method that generates the simulated activity data and computes statistical measures for the data is then called by the activity Simulation Engine method. Lastly, the application calls the display Data function to show the outcomes. Depending on whether the data is discrete or continuous, the activity Simulation Engine method calls the discrete Event Simulation and continuous Event Simulation methods to construct the simulated activity data. These techniques take the input data and utilize a random number generator to produce values that have a normal distribution with mean and standard deviation values that are given. The software generates the simulated data and then computes a number of statistical measures, such as the mean value, the standard deviation, and the total value for each event. The display HashMap Data method is used to display the measures that have been saved in hash maps. All things considered, this program offers a helpful tool for simulating and analyzing server activity data, which may be utilized to spot possible anomalies or areas of concern.

The application has been modified in response to user feedback and new opportunities. In conclusion, our user-friendly web scraping tool for job listings provides a good deal of capability, providing users with a reliable, efficient, and ethical tool to collect, analyze, and present relevant job market data.

V. Proposed Methodology

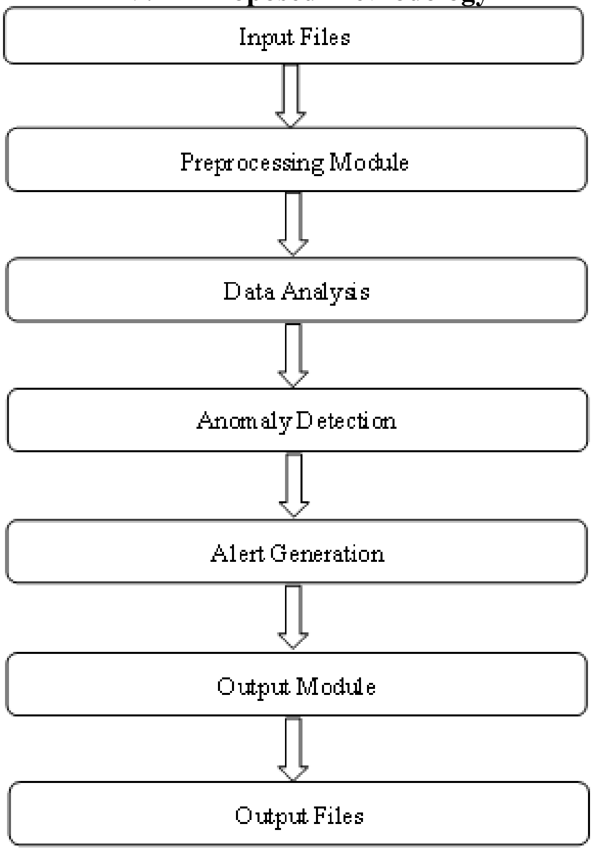


Fig: Execution of coding Modules.

This Java project's intrusion detection system design often consists of several interconnected components that work together to track and identify anomalous activity.

Files Accepted: This module accepts the stats.txt and event.txt files as input.

Preprocessing Module: This module preprocesses, cleans, and changes the incoming data in order to make it ready for analysis. It can also be used for feature extraction and selection.

Data Analysis: This module analyzes the preprocessed data to extract essential information such as trends and patterns. It may use machine learning algorithms, statistical techniques, or other methods to identify patterns, relationships, and abnormalities.

Anomaly Detection: This module uses the output from the data analysis module to search for abnormalities or potential intrusions. It may utilize statistical models, rule-based systems, or machine learning techniques to identify deviations from expected behavior.

Alert Generation: This module notifies users when it detects anomalies or potential intrusions. It may use heuristics, thresholds, or other criteria to determine when to send out an alert.

The output module is in charge of presenting and formatting the alert data so that it is suitable for the final user. It may include dashboards, reports, or other visual aids to help the user comprehend the warnings and take the appropriate action.

Output Files: This module copies the output data to the output files for further analysis or auditing.

VI. Architecture

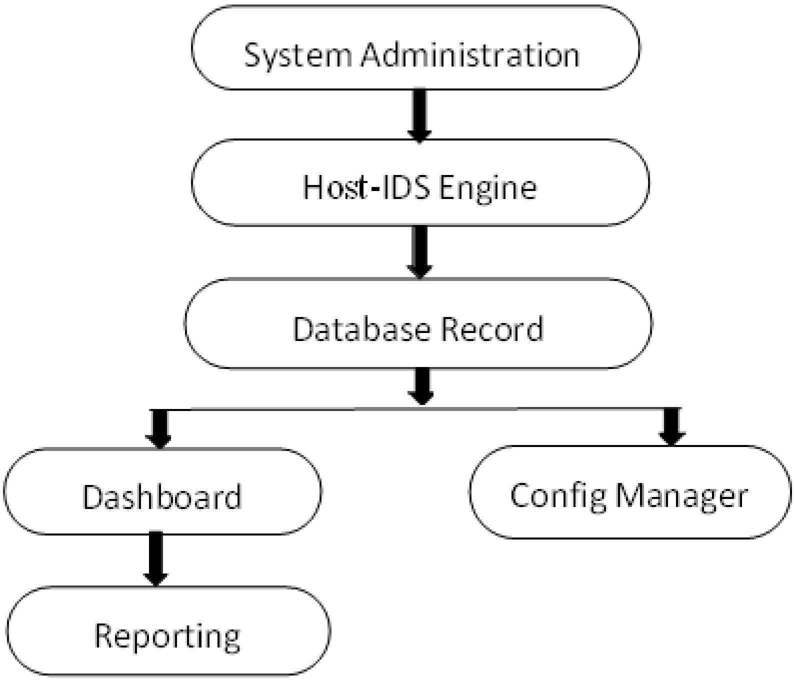


Fig: Architecture of project.

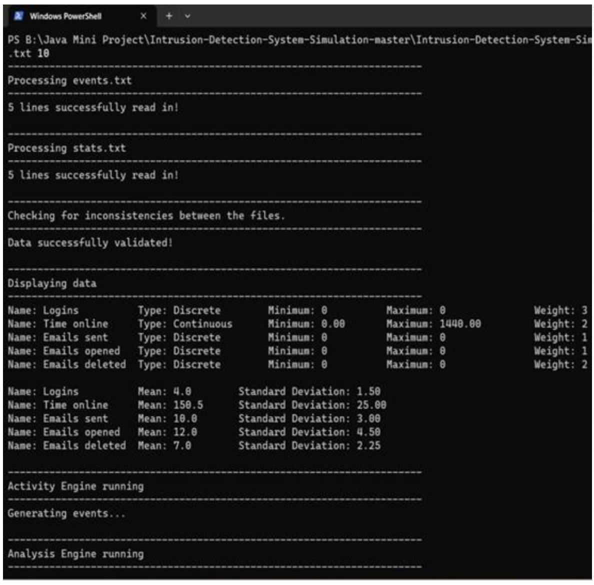
VII. Conclusion

A Database Intrusion Detection System (IDS) project is a crucial instrument for safeguarding sensitive data against online attacks in enterprises. Data collection, anomaly detection, signature-based detection, real- time alerting, forensic analysis, user activity monitoring, adjustable policy enforcement, reporting, and scalability are just a few of the features and capabilities that the project can offer in order to identify and react to attacks.

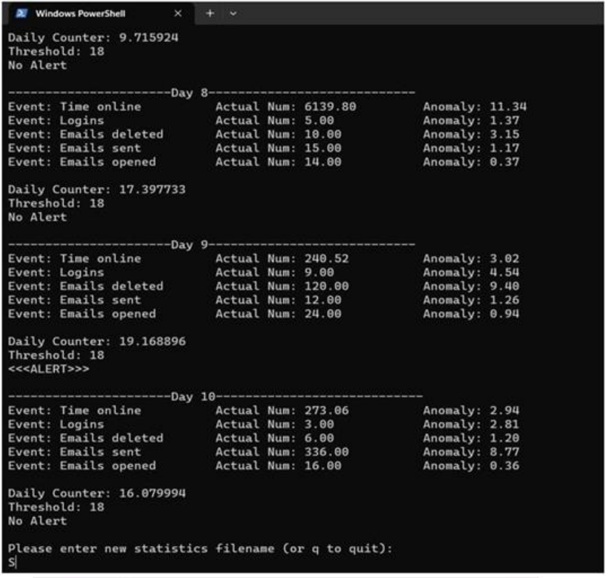
The prospective development areas for a database IDS project include machine learning, cloud-based deployment, threat intelligence integration, automated response, analytics on user and entity activity, and integration with SIEM. The project's future scope is equally promising. These features enable database IDS to keep up with changing cyberthreats and carry on offering reliable organizational data security.

Organizations may enhance their overall security posture and preserve data confidentiality, integrity, and availability with the support of a well-designed database intrusion detection system (IDS). It is a crucial tool for every business or industry that uses databases to handle and keep sensitive data. Last but not least, the HOST-based intrusion detection system (IDS) in this project focuses on protecting sensitive financial information belonging to any industry or organization, thereby making a significant contribution to the fields of computational intelligence and cyber security. The Host-Based Intrusion Detection System in a Bank Database, in conclusion The Java project is a software programme made to keep an eye on and identify possible security risks in the bank's database. It finds suspicious activity using a variety of log formats and detection rules, and when needed, it sends out notifications. The project include generating a set of software requirements, designing and implementing a system architecture, and producing a variety of models and diagrams to aid in the development process. The project's overarching objective is to strengthen the bank's database's security and guard against any potential invasions.

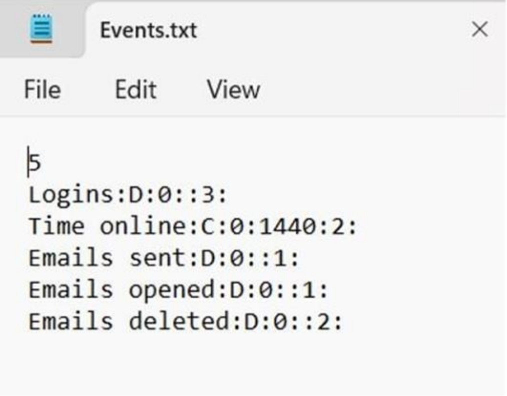
VIII. Results And Output



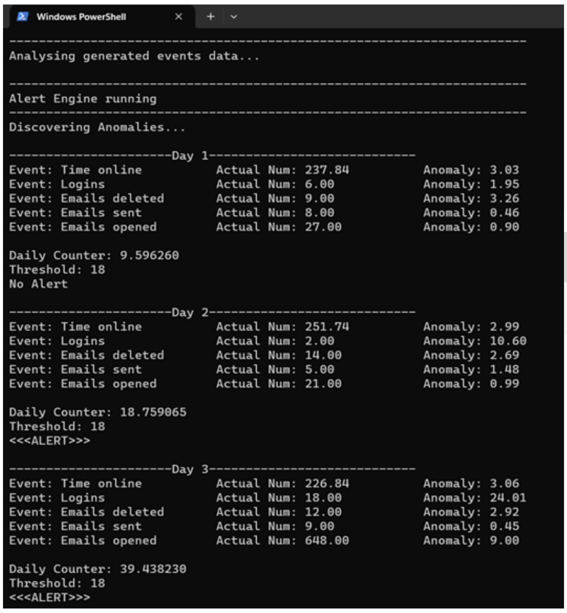
```
PS B:\Java Mini Project\Intrusion-Detection-System-Simulation-master\Intrusion-Detection-System-Sim
.txt 10
----------------------------------------------------------------------
Processing events.txt
----------------------------------------------------------------------
5 lines successfully read in!

----------------------------------------------------------------------
Processing stats.txt
----------------------------------------------------------------------
5 lines successfully read in!

----------------------------------------------------------------------
Checking for inconsistencies between the files.
----------------------------------------------------------------------
Data successfully validated!

----------------------------------------------------------------------
Displaying data
----------------------------------------------------------------------
Name: Logins         Type: Discrete       Minimum: 0         Maximum: 0              Weight: 3
Name: Time online    Type: Continuous     Minimum: 0.00      Maximum: 1440.00        Weight: 2
Name: Emails sent    Type: Discrete       Minimum: 0         Maximum: 0              Weight: 1
Name: Emails opened  Type: Discrete       Minimum: 0         Maximum: 0              Weight: 1
Name: Emails deleted Type: Discrete       Minimum: 0         Maximum: 0              Weight: 2

Name: Logins         Mean: 4.0      Standard Deviation: 1.50
Name: Time online    Mean: 150.5    Standard Deviation: 25.00
Name: Emails sent    Mean: 10.0     Standard Deviation: 3.00
Name: Emails opened  Mean: 12.0     Standard Deviation: 4.50
Name: Emails deleted Mean: 7.0      Standard Deviation: 2.25

----------------------------------------------------------------------
Activity Engine running
----------------------------------------------------------------------
Generating events...

----------------------------------------------------------------------
Analysis Engine running
----------------------------------------------------------------------
```



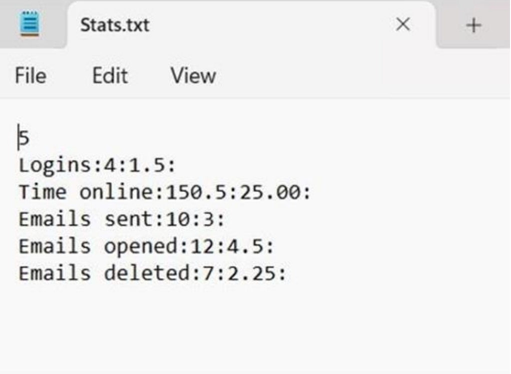
```
Daily Counter: 9.715924
Threshold: 18
No Alert

---------------------Day 8-----------------------------
Event: Time online          Actual Num: 6139.80         Anomaly: 11.34
Event: Logins               Actual Num: 5.00            Anomaly: 1.37
Event: Emails deleted       Actual Num: 10.00           Anomaly: 3.15
Event: Emails sent          Actual Num: 15.00           Anomaly: 1.17
Event: Emails opened        Actual Num: 14.00           Anomaly: 0.37

Daily Counter: 17.397733
Threshold: 18
No Alert

---------------------Day 9-----------------------------
Event: Time online          Actual Num: 240.52          Anomaly: 3.02
Event: Logins               Actual Num: 9.00            Anomaly: 4.54
Event: Emails deleted       Actual Num: 120.00          Anomaly: 9.40
Event: Emails sent          Actual Num: 12.00           Anomaly: 1.26
Event: Emails opened        Actual Num: 24.00           Anomaly: 0.94

Daily Counter: 19.168896
Threshold: 18
<<<ALERT>>>

---------------------Day 10-----------------------------
Event: Time online          Actual Num: 273.06          Anomaly: 2.94
Event: Logins               Actual Num: 3.00            Anomaly: 2.81
Event: Emails deleted       Actual Num: 6.00            Anomaly: 1.20
Event: Emails sent          Actual Num: 336.00          Anomaly: 8.77
Event: Emails opened        Actual Num: 16.00           Anomaly: 0.36

Daily Counter: 16.079994
Threshold: 18
No Alert

Please enter new statistics filename (or q to quit):
S
```



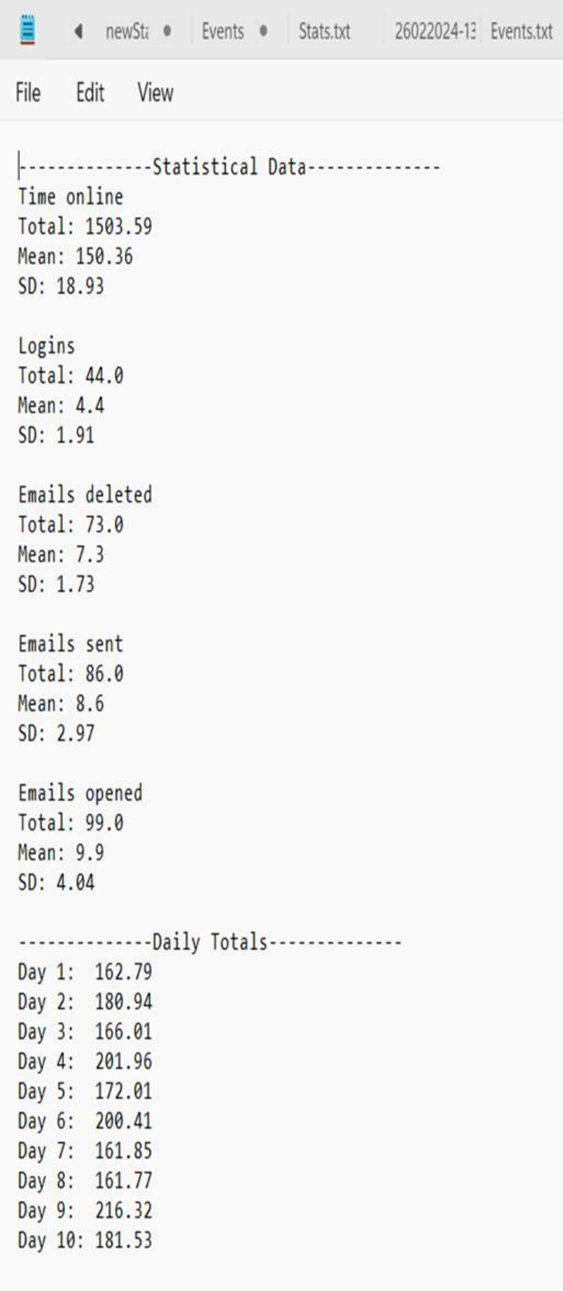
```
5
Logins:D:0::3:
Time online:C:0:1440:2:
Emails sent:D:0::1:
Emails opened:D:0::1:
Emails deleted:D:0::2:
```



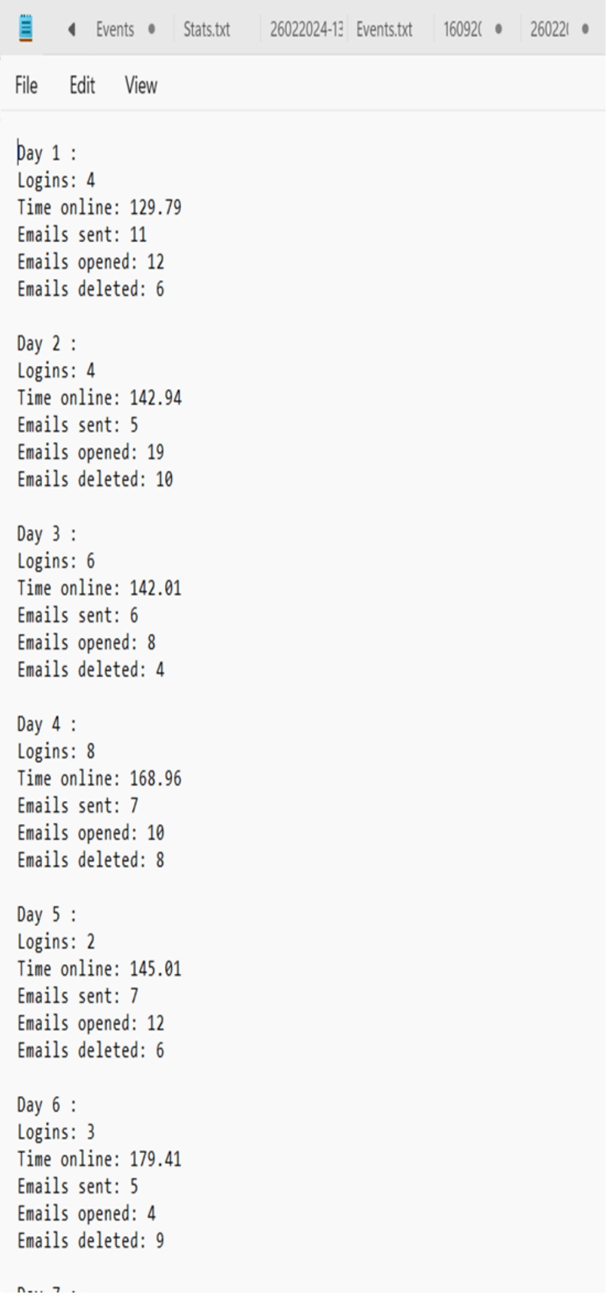
```
----------------------------------------------------------------------
Analysing generated events data...

----------------------------------------------------------------------
Alert Engine running
----------------------------------------------------------------------
Discovering Anomalies...

---------------------Day 1-----------------------------
Event: Time online          Actual Num: 237.84          Anomaly: 3.03
Event: Logins               Actual Num: 6.00            Anomaly: 1.95
Event: Emails deleted       Actual Num: 9.00            Anomaly: 3.26
Event: Emails sent          Actual Num: 8.00            Anomaly: 0.46
Event: Emails opened        Actual Num: 27.00           Anomaly: 0.90

Daily Counter: 9.596260
Threshold: 18
No Alert

---------------------Day 2-----------------------------
Event: Time online          Actual Num: 251.74          Anomaly: 2.99
Event: Logins               Actual Num: 2.00            Anomaly: 10.60
Event: Emails deleted       Actual Num: 14.00           Anomaly: 2.69
Event: Emails sent          Actual Num: 5.00            Anomaly: 1.48
Event: Emails opened        Actual Num: 21.00           Anomaly: 0.99

Daily Counter: 18.759065
Threshold: 18
<<<ALERT>>>

---------------------Day 3-----------------------------
Event: Time online          Actual Num: 226.84          Anomaly: 3.06
Event: Logins               Actual Num: 18.00           Anomaly: 24.01
Event: Emails deleted       Actual Num: 12.00           Anomaly: 2.92
Event: Emails sent          Actual Num: 9.00            Anomaly: 0.45
Event: Emails opened        Actual Num: 648.00          Anomaly: 9.00

Daily Counter: 39.438230
Threshold: 18
<<<ALERT>>>
```



```
5
Logins:4:1.5:
Time online:150.5:25.00:
Emails sent:10:3:
Emails opened:12:4.5:
Emails deleted:7:2.25:
```



```
-------------Statistical Data-------------
Time online
Total: 1503.59
Mean: 150.36
SD: 18.93

Logins
Total: 44.0
Mean: 4.4
SD: 1.91

Emails deleted
Total: 73.0
Mean: 7.3
SD: 1.73

Emails sent
Total: 86.0
Mean: 8.6
SD: 2.97

Emails opened
Total: 99.0
Mean: 9.9
SD: 4.04

-------------Daily Totals-------------
Day 1:  162.79
Day 2:  180.94
Day 3:  166.01
Day 4:  201.96
Day 5:  172.01
Day 6:  200.41
Day 7:  161.85
Day 8:  161.77
Day 9:  216.32
Day 10: 181.53
```



```
Day 1 :
Logins: 4
Time online: 129.79
Emails sent: 11
Emails opened: 12
Emails deleted: 6

Day 2 :
Logins: 4
Time online: 142.94
Emails sent: 5
Emails opened: 19
Emails deleted: 10

Day 3 :
Logins: 6
Time online: 142.01
Emails sent: 6
Emails opened: 8
Emails deleted: 4

Day 4 :
Logins: 8
Time online: 168.96
Emails sent: 7
Emails opened: 10
Emails deleted: 8

Day 5 :
Logins: 2
Time online: 145.01
Emails sent: 7
Emails opened: 12
Emails deleted: 6

Day 6 :
Logins: 3
Time online: 179.41
Emails sent: 5
Emails opened: 4
Emails deleted: 9
```

References

[1] A Survey On Anomaly Detection Was Conducted By Banerjee, Kumar, And Chandola In 2009 And Published In Acm Computing Surveys, Vol. 41, No. 3, Pp. 1-58.

[2] In 2018, The International Journal Of Computer Applications Published A Comprehensive Assessment Of An Intrusion Detection System, Titled "Intrusion Detection System: A Comprehensive Review," Written By Mohammed Alkhazraji, Alkhazraji, And Abdulkareem, A. M.

[3] In 2017, Sahu And Reddy Conducted A Thorough Analysis Of Anomaly-Based Intrusion Detection Systems In The Journal Of Network And Computer Applications, Volume 94, Pages 62–82.

[4] "Review Of Anomaly Detection Techniques For Intrusion Detection System," A. L. N. Reddy And D. D. K. R. Chowdary, International Journal Of Engineering Research And Applications, Vol. 6, No. 8, Pp. 58-63, 2016.

[5] "Intrusion Detection System: A Review," S. K. Jha And R. K. Pandey, International Journal Of Computer Applications, Vol. 64, No. 4, Pp. 1-7, 2013.

[6] Ramaswamy, N., & Rajagopalan, M. R. (2012). A Survey Of Database Intrusion Detection Techniques. Journal Of Network And Computer Applications, 35(3), 1017-1037.

[7] Xu, J., & Chen, Y. (2019). A Lightweight Database Intrusion Detection System Based On Convolutional Neural Network. Information Sciences, 484, 122-135.

[8] Fu, Y., & Sun, Z. (2018). Database Intrusion Detection System Based On Fuzzy Association Rules. Journal Of Ambient Intelligence And Humanized Computing, 9(6), 2029-2037.