Journal of Engineering Sciences
Vol 11, Issue 6,June/2020
ISSN NO:0377-9254

ENHANCING MAIL SERVER SECURITY USING MACHINE LEARNING

*Swati S Maddur[1], Richul N Prasad[2], P Jyothi Priya[3], Pruthvi Sainath Reddy[4] and Sumithra Devi K.A[5]

[1][2][3][4] *BE Students, Department of Information Science and Engineering*

[5]*HOD, Department of Information Science and Engineering*

[1][2][3][4][5] *Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India*

swatism312@gmail.com[1],richulnprasad@gmail.com[2], jyothi.priya2710@gmail.com[3], sainathreddy98@gmail.com[4], deanacademics@dsatm.edu.in[5]

ABSTRACT

In today's digital world, most organizations and users who are concerned about the privacy and security of their e-mail transactions use their own mail servers. Even with the vast technological advancements, mail server security is limited to password-based authentication and a fixed set of rules. A malicious user with intention can impersonate a naive user by misleading such rule-based security models. In this paper, we propose an adaptive self-learning model that uses the capabilities of machine learning to record each user's usage patterns independently based on the logs generated by their own email transactions. Our model is trained over the logs that are native to one's mail server in opposition to the third-party datasets where the ethicality and authenticity are uncertain and hence the model inches towards being truly secure. The metadata in every mail log allows us to extract crucial information like date, time, username, IP address, ISP, Geo-Location, etc. This processed information correlates to specific usage behaviour which can be plotted onto a machine. Our model uses principal component analysis for dimensionality reduction and further the anomaly detection algorithm using the average K-NN method to perform clustering. Through clustering, the model identifies the anomalous changes in usage patterns and flag them as suspicious for further suitable action.

Keywords: *Mail Server Security, IMAP, SMTP, Anomaly Detection, Machine Learning, Average K-NN, PCA, GeoIP2.*

1. INTRODUCTION

With the advent of the information age, e-mail has become an indispensable means of communication because of its convenience and speed. Users send emails frequently resulting in the mail server generating an enormous number of logs. These logs contain a lot of valuable information [1]. They record people's username, remote IP address, local IP

The effective analysis and processing of the mail logs is a crucial task for the working and management of the mail system. The mail server generates a large amount of data every day. Most of the mail logs (such as smtpd, pop3, etc.) are not only large in data size, they also look obscure[1]. In the face of the discovery of mail anomalies and the need to check the mail delivery status, only relying on the manual work of the administrator to view the log records each time a message is queried which takes a minute or two makes the system highly prone to human error. Larger the workload, the more inefficient and error prone is the result.

The default protocol that is used to send and receive e-mail is Simple Mail Transfer Protocol (SMTP) while the Internet Mail Access Protocol (IMAP) by defaults is used to access the e-mails. The most significant security weakness in IMAP protocol is that it stores e-mails on the server as plaintext and also allows plain login authentication from remote servers which makes it more vulnerable. IMAP implementation exposes all users thereby permitting remote users to authenticate themselves with plaintext user ID and password. The lack of support for strong authentication, like multifactor authentication, is probably the most intransigent challenge to defenders[2]. Since the IMAP protocol alone leaves the mails exposed on the server, the mail server's security requires protection from spoofing attacks.

For traditional standalone systems, administrators can investigate the system logs manually and detect anomalies based on a keyword search or regular string matching. However, such methods that rely heavily on manual inspection have become inadequate for large scale systems. Most existing detection methods require a prior knowledge (third-

party datasets); they use the patterns of known attacks to identify anomalies; thus, these methods cannot address the new or unknown threats[3]. Furthermore, the increasing scale and complexity of modern systems make the volume of logs increase quickly, which requires that the newly developed anomaly detection methods should also have a accuracy and low computational complexity. Facing the above challenges, we propose a new self learning, independent and adaptable machine learning model to detect suspicious users.

The distance based outlier algorithms uses the distance between the data points. The distance of the data points with its neighbouring data point is calculated and checked, if it is close then it is considered as inlier and if the distance is far then it is consider as outlier[4]. The traces of malicious activities is reflected in the data, careful analysis of this data reveals the presence of the suspicious user. The anomaly traces in the data have common characteristics which makes anomaly detection possible[5]. Hence suggesting that the anomaly detection algorithms are the right fit to train our model.

To achieve the goal of anomaly detection, proper features need to be extracted to characterize the user behaviour. The session information of the users can be used to detect potential abnormal behaviours, which can minimize the computational complexity. Features related to logging activity are mainly used to characterize the users logging behaviors', to check whether the user login to a server is from a usual host, to determine whether the user login occurs during the usual period of time or in a usual location, etc[12]. These features are used to describe the logging behaviour characteristics from the 3W (Who, Where, and When) aspects[3].

2. PROBLEM STATEMENT

The existing legacy systems of mail server security does not go beyond password-based authentication. Most of us use passwords to access our mails which provides only a layer of security[7]. If the password is exposed to a malicious user, he/she can pose to be the actual user or even extract the user sensitive information. Most mail server security models these days make use of security policies based on a fixed set of rules[13]. At the same time, another aspect of this is the regular use of existing machine learning models, which is hugely dependent on the third-party datasets. There are various questions raised about the ethicality of how such data are obtained and also the authenticity of the data from third-parties. Such data on its own could be a breach of somebody's privacy. Likewise, the data could be manipulated such that the end result of the machine learning model is hampered.

To overcome the above challenges, we propose a machine learning anomaly detection model. The model will train itself from Zero-Data for every instance individually over the logs generated by one's mail server. The model will gradually pick up the usage patterns and behaviour of a user and using the anomaly detection model, it will further flag any suspicious unusual behaviour from any user that could lead us to potentially blocking threats by considering them as unauthenticated. The model will entirely run on a local network in conjugation with the mail server over the logs generated which are also native to the mail server, hence the data never leaves one's trusted system. The model provides the user with the enhanced security of his mail server and at the same time respects, one's civil liberty rights by not collecting any data out of his own system.

3. LITERATURE SURVEY

Mail server these days are exposed to a whole lot of threats [2][6] but the lack of an efficient authentication system is the most alarming considering a password is among the easiest to gain access of [7]. Every mail server generates logs over each transaction of e-mail. These logs provide us with a lot of essential metadata that can be utilized to analyze the behaviour of a user. The mail server is up and running always which leads to generation of enormous amount of logs daily. The manual evaluation of suspicious behaviour by the administrator makes the mail server prone to human error while also being time consuming[3].

Before we can utilize these logs in a machine learning algorithm we need to process these logs containing raw metadata and extract useful information from it. Thus under log processing, the crucial information like IP address, date and time present in the logs can be used to further extract other essential information like geolocation, ISP and 'how many days ago the mail was sent' using the databases like Maxmind's GeoIP2Lite database [8][14] and also DateTime module respectively. This information extracted can now be used to study the general usage patterns of a user . The model most resourceful to our problem statement is

a distance-based anomaly detection model[4] which can clusterize the usage pattern of a user into one or more clusters and output any out-of-cluster activity as anomalies or outliers which gives rise to a suspicion that a user's account might potentially be compromised.

4. METHODOLOGY

The proposed model is composed of three main phases, which are:

- Mail Server Setup
- Pre-processing of Generated Logs
- Deploying on a Machine Learning Model

A. Mail Server Setup

In order to build a mail server security model, the first3 step is to deploy a mail server itself. For this a linux based operating system called debian is used. It makes use of two main application, namely postfix and dovecot. Postfix is not only free but also an open-source Mail Transfer Agent(MTA) which routes and delivers electronic mail. Dovecot is also an open source Mail Delivery Agent(MDA) which helps user access mails stored in the server.

Prior to setting up postfix (SMTP server) we require a domain name for the mail server. For example, consider a mail id mymail@somedomain.com in which 'mymail' is the host name and 'somedomain.com' is the domain name. The host name along with our IP address is inserted in the 'hosts' file in system to ensure that the postfix recognizes the mail server by the corresponding domain name. In the main configuration file (/etc/postfix/main.conf) of postfix we need to ensure that the following directives are set as follows to achieve right working of the postfix:

- inet_interfaces = loopback-only
- mydestination = domain_name

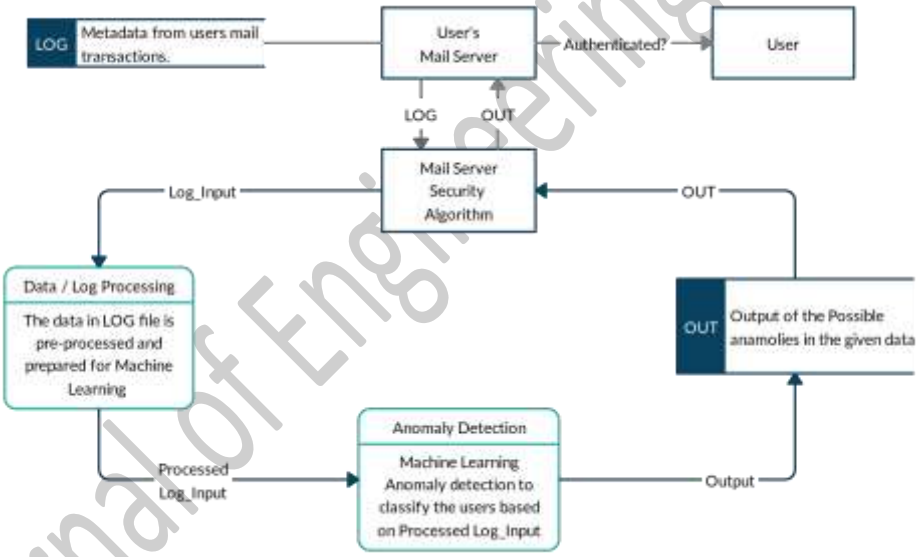

Fig 1: Dataflow Diagram for Proposed Model

Setting the inet_interfaces to loopback-only ensures that connections can originate from the machine on which mail server is deployed while mydestination is set to domain name chosen previously.

In a similar way the dovecot (IMAP server) is setup to enable accessing of emails on our own server. Dovecot has four configuration files, each carrying its own responsibility:

- /etc/dovecot/dovecot.conf
- /etc/dovecot/conf.d/10-auth.conf
- /etc/dovecot/conf.d/10-master.conf
- /etc/dovecot/conf.d/10-mail.conf

After Dovecot has been setup in the system, we have to configure the file '/etc/dovecot/dovecot.conf' and look for the line 'listen = *, :' where * and : indicates that dovecot will listen to both IPv4 and IPv6 interfaces respectively. Next, the file '/etc/dovecot/conf.d/10-auth.conf' is edited to ensure enabling of the authentication by plain login method. the file '/etc/dovecot/conf.d/10-mail.conf' is opened, and the mail location is to be changed from mbox format to Maildir for convenience. Finally, the file '/etc/dovecot/conf.d/10-master.conf' is opened. The

smtp-auth block is to be searched for and add permission rights so that postfix can communicate with dovecot internally.

Having setup postfix and dovecot, our mail server can now be used to send and receive mails using any mail client such as thunderbird. The transaction of mails on any given mail server generates logs which are nothing but the metadata of the email transaction. These logs are dumped into mail.log file[15]. The logs concerning us are those of IMAP and POP3 server and each imap-login log consists of data such as date, time, username, rip, lip, session id and so on. But these are just raw logs which cannot be directly fed[1] to the machine learning model. Hence we introduce the concept of pre-processing the logs to structure them.

B. Pre-processing of Generated Logs

Log processing is one of the most crucial stage where the log for each instance is put into a structured format such that a key value representation is available for each information in the logs[1]. This phase that first focuses on structuring the data into key value pairs using the power of dataframes and other data structures of the python programming language.

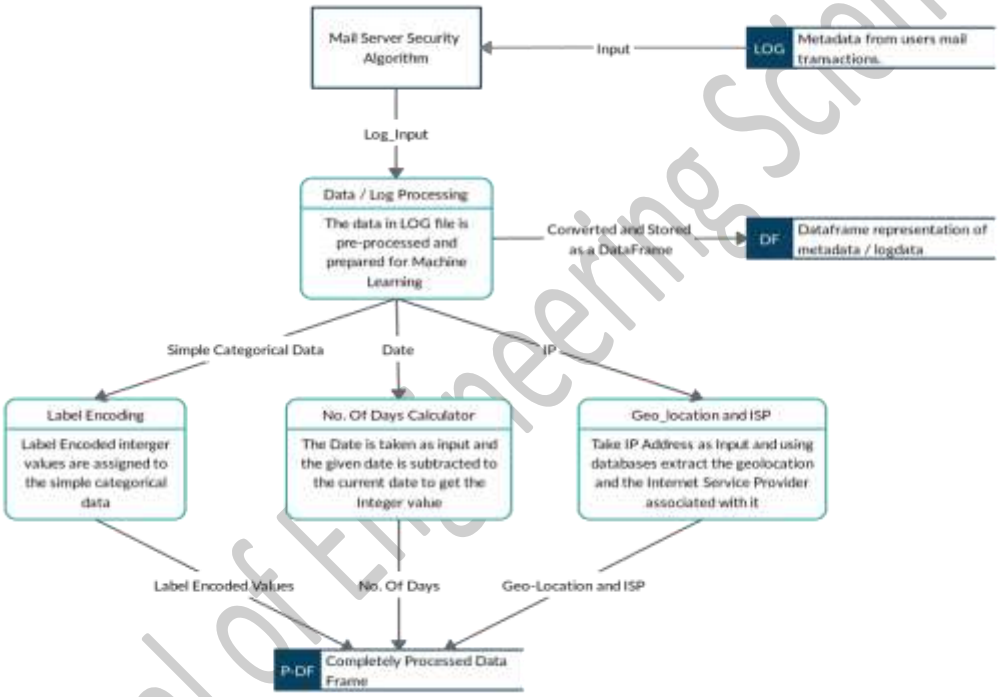

Fig 2: Dataflow Diagram for Log Processing

Each log has normally thousands of lines in them[9], and each line has standard essential data that describe the particular transaction , and hence can be used to come to effective conclusions.

Regex expression is made use of to filter out any unwanted lines in the log. This filtered log data is now tokenized and stored in a dictionary, which can then be converted to a dataframe directly. The log consists of IP address which on its own is capable of providing tons of information about the user it is referring to. Hence, we make use of the IP address to extract sender's location[14] and ISP(Internet Service Provider). The user's location from which he sends the mail is generally specific to a few common locations, and any location that pops the first time can be flagged for further consideration. The ISP of the internet service that user uses is again a commonly constant information, and any unusual variations in the same as well needs to be flagged for further considerations. We make use of the Maxmind's GeoIP2Lite database which helps us extract the sender's city and country using the IP address. In order to extract ISP we make use of the IPWhois package which again gives a lot of information regarding the IP but we are only concerned with 'asn_description' section which gives us the sender's ISP.

The IP, User Name, ISP information are all categorical data , in simple words, data formats such as strings cannot be fed directly into the machine learning models. Hence, sklearn's label encoder can be used to effectively encode the data into numerical data. The Date and Country information are both special cases that need to be treated separately. The date and country information needs to be fed intelligently to preserve the meaning of the data. Encoding will assign a label to the same, but we cannot use the information effectively when done so. For date, we use an algorithm that calculates the no. of days between the present date and the date that we are handling. For country we use Longitude and Latitude information to mathematically calculate the effective distance between the sender's location to the location of where our server is located.

Finally, we use a different file for the fitting part, this is with the intent to keep the current processing consistent with all the previously generated logs. Hence, we fit the model using the file that has the entire log dump and the unique data in that is fitted into the model and thenencode the existing file. The final part is the feature selection which is done manually by judging what features we need essentially i.e. Username, Remote IP, Local IP, Date, ISP, Distance. The generated dataframe is the final completely processed log data that is ready to be fit into a machine learning model without raising any errors and exceptions.

C. Deploying on a Machine Learning Model

One of the most important decision to be taken while deploying a machine learning model is the need to choose the most appropriate algorithm or model that can provide us with the most accurate results possible. Having tested multiple techniques such as naïve bayes, mean shift clustering, Gaussian distribution and so on, we came across the concept of anomaly detection[5][10] using outliers which, in our search for algorithms, turned out to be most appropriate algorithm to provide accurate results.

An outlier[4] is any data point which differs greatly from the rest of the observations in a dataset. Although we have multiple features in our dataset,

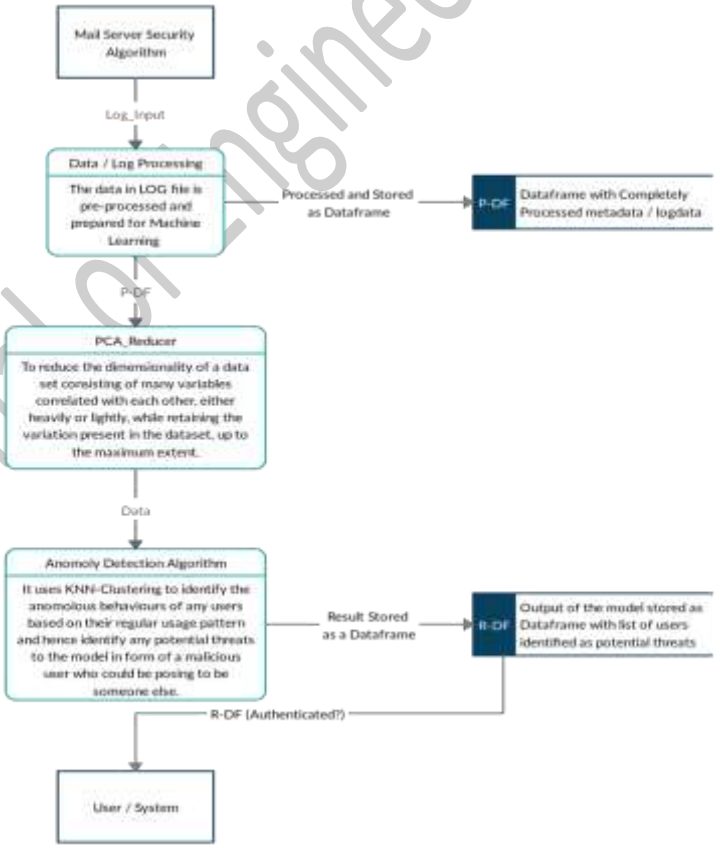

Fig 3: Dataflow Diagram of Anomaly Detection

we reduce it to two features by feature extraction method PCA to reduce the complexity. After a certain point, more features or dimensions can decrease a model's accuracy since there is more data that needs to be generalized — this is known as the curse of dimensionality[18]**.** Dimensionality reduction is way to reduce the complexity of a model and avoid over fitting. We use feature extraction to derive information from the feature set to construct a new feature subspace using principal component analysis (PCA) to compress our dataset onto a lower- dimensional feature subspace (two features) with the goal of maintaining most of the relevant information[18].

After dimensionality reduction, the dataframe is added with new columns called 'dependency state' that predict the dependency of each column in getting the most accurate results. The values in this column range from positive to negative values but make it tough to plot graph and view datapoints. Hence, we re-scale the data using scikit-learn with the help of MinMaxScaler class where the reduced data is reshaped between the range 0 to 1.

The dataframe is now finally ready to be fed to the machine learning model. The dataset is fit into the anomaly detection model that uses average K-NN clustering and predicts the total number of outliers and inliers in the given dataset. Average K-NN algorithm is preferred over the standard K-NN[11] because in standard algorithm multiple clusters are formed and as more number of logs are obtained there could be chances of the data points being considered into the wrong cluster. Whereas average K-NN outlier detection provides only two clusters namely, outliers and inliers. The average K-NN method is a distance based outlier detection approach that calculates the distance between datapoints and its distance from the centroid of each cluster thus, distinguishing authenticated users from unauthenticated users.

5. RESULTS

For the purpose of experiment, we collected two types of mail logs, namely IMAP and POP3 logs. Table II shows detailed information of the log format. These logs are firstly processed where the essential information like ISP, geolocation and so on are extracted from the crucial information present in the logs post which simple categorical data such as user, rip, lip, isp etc are label encoded. For special cases like date, an algorithm to calculate the number of days between current date and given date is used whereas for country, longitude and latitude information is used to mathematically calculate the effective distance between the sender's location and the location where our server is located. Table I given below displays the encoded output.

Table I: Encoded Output

	user	rip	lip	date_time	isp	distance
0	1	5	0	56	4	1234.16
1	2	3	0	56	2	0
2	2	3	0	56	2	0
3	2	3	0	56	2	0
...	...	...	...	...	...	...
21	6	3	0	56	2	0
22	1	7	0	56	4	1234.16
23	4	3	0	56	2	0
24	4	3	0	56	2	0

The encoded output in Table I consist of multiple features and hence principal component analysis (PCA) is used to reduce its dimensionality while also retaining important data. This reduced data is fed to the anomaly detection model.

The average K-NN anomaly detection model is used which can clusterize the usage pattern of a user into one or more clusters and output any out-of-cluster activity as anomalies[11] or outliers which gives rise to a suspicion that a user's account might potentially be compromised

Table II: IMAP Logs

ID	Log
1	Feb 27 14:35:50 imap-login: Info: Disconnected user = <MTY5WG9OUXR5aENTLgo> method=PLAIN, rip=42.106.46.149, lip=172.18.0.2, session=<bcjnBouf/Fwqai6V>
2	Feb 27 14:36:03 imap-login: Info: Disconnected user=<MXMvQVN0OFVPMDF2Lgo>, method=PLAIN, rip=172.18.0.8, lip=172.18.0.2, session=<GEexB4ufhoGsEgAI>
.	…

Table III: Final Output

OUTLIERS: 4 INLIERS: 21

	date	time	info	...	isp	authenticated?
8	FEB 27	14:46:31	Disconnected	...	RELIANCEJIO	1
9	FEB 27	14:46:40	Disconnected	...	RELIANCEJIO	1
11	FEB 27	14:49:39	Disconnected	...	BHARATI	1
18	FEB 27	14:56:11	Disconnected	...	RELIANCEJIO	1

Table III displays the final output obtained which distinguishes suspicious users from normal users with the aid of logs. If the value in last column 'authenticated?' is '1' then the log is unauthenticated, otherwise it is considered as authenticated.

For any anomaly detection model to be considered, initially it should generate a small number of outliers to begin with. When the same was tested for various algorithms such as HBOS, isolation forest, K-NN and average K-NN, out of the four average K-NN algorithm was the only one to generate less number of outliers initially.

Fig 4 represents bar graph plotted to visualize the false alarm rate generated by each algorithm. The graph takes algorithms implemented on x-axis and number of outliers on the y-axis. As the number of logs generated by a mail server increase, the load on model may also increase. Hence it is of atmost importance to choose an algorithm which can handle enormous amount of logs.

Fig 5 represents a line-graph with dataset range on x-axis and number of outliers on y-axis. While the three algorithms keep on increasing number of outliers with increase in dataset range, average K-NN model works on predicting precise number of outliers as shown in the fig 5.

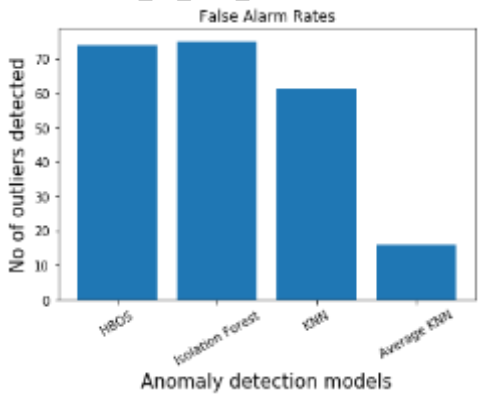

Fig 4: Bar Graph for False Alarm Rates

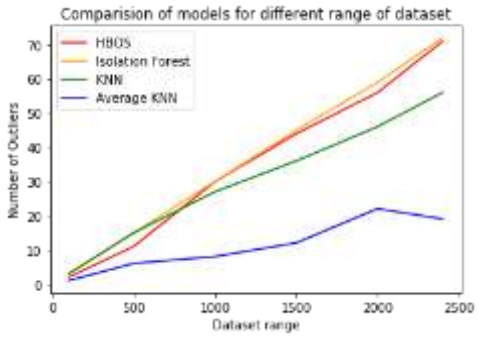

Fig 5: Precision Comparision for Various Models

The graphs in fig 4 and 5 show that the proposed average K-NN model performs much better when compared to the other three models. The model is capable of predicting the most precise output and shows promising behaviour when it comes to handling a huge amount of data.

6. CONCLUSION

The proposed model introduces an advanced approach to email authentication. The model effectively uses the logs generated by the mail server and studies the email behavior as to regular patterns of usage for every user on the mail server and utilizes the Anomaly detection model to track unusual changes in the above-recorded email behavior. This provides us with a much efficient and modern-day approach towards mail server security where access to one's compromised password is not the only parameter that leverages a malicious user into posing as somebody else. The model enhances the security of the mail server by increasing the hurdles against any malicious or treacherous intentions of hackers.

REFERENCES

[1] Yun, B. (2018, May). Mail Scheme Log Processing Based on ELK. In 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018). Atlantis Press.

[2] Shitole, H. P., & Divekar, S. Y. (2019). Secure Email Software using e-SMTP.
[3] Liu, Z., Qin, T., Guan, X., Jiang, H., & Wang, C. (2018). An integrated method for anomaly detection from massive system logs. IEEE Access, 6, 30602-30611.
[4] Mandhare, H. C., & Idate, S. R. (2017, June). A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 931-935). IEEE.
[5] Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018, April). A survey on anomaly based host intrusion detection system. In Journal of Physics: Conference Series (Vol. 1000, No. 1, p. 012049). IOP Publishing.
[6] Zeng, Y. G. (2017, June). Identifying email threats using predictive analysis. In 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security) (pp. 1-2). IEEE.
[7] Herley, C., & Schechter, S. (2018). Distinguishing Attacks from Legitimate Traffic at an Authentication Server. Microsoft, Technical Report MSRTR-2018-19.
[8] Khedr, W. I., Emara, A. G., & Ziedan, I. (2018). A Treemap Based Network Visualization Scheme For Detecting Network Attacks. Journal of Theoretical & Applied Information Technology 96(1).
[9] Dymshits, M., Myara, B., & Tolpin, D. (2017, October). Process monitoring on sequences of system call count vectors. In 2017 International Carnahan Conference on Security Technology (ICCST) (pp. 1-5). IEEE.
[10] Wang, H., Bah, M. J., & Hammad, M. (2019). Progress in Outlier Detection Techniques: A Survey. IEEE Access, 7, 107964-108000.
[11] Djenouri, Y., Belhadi, A., Lin, J. C. W., & Cano, A. (2019). Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow. IEEE Access, 7, 10015-10027.
[12] T. Li, A. Mehta and P. Yang, "Security Analysis of Email Systems," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, 2017, pp. 91-96.
[13] Kumar, P., & Iqbal, F. (2019, April). Credit Card Fraud Identification Using Machine Learning Approaches. In 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT) (pp. 1-4). IEEE.
[14] Komosný, D., Vozňák, M., & Rehman, S. U. (2017). Location accuracy of commercial IP address geolocation databases.
[15] Bao, L., Li, Q., Lu, P., Lu, J., Ruan, T., & Zhang, K. (2018). Execution anomaly detection in large-scale systems through console log analysis. Journal of Systems and Software, 143, 172-186.
[16] Souri, A., & Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. Human-centric Computing and Information Sciences, 8(1), 3.
[17] Patra, S., Naveen, N. C., & Prabhakar, O. (2016, May). An automated approach for mitigating server security issues. In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 1075-1079). IEEE.
[18]G. T. Reddy et al., "Analysis of Dimensionality Reduction Techniques on Big Data," in IEEE Access, vol. 8, pp. 54776-54788, 2020.