Fuzzy Systems and Soft Computing
ISSN: 1819-4362

Machine Learning-Based Detection and Mitigation of Privilege Escalation Attacks in Cloud Environments

Mr.Amarthaluri Manoj Kumar
Dept. of CSE(AI & ML)
S R Gudlavalleru Engineering College
Gudlavalleru, India
manojiceu@gmail.com

Mr.Doddapaneni Hemanth
Dept. of CSE(AI & ML)
S R Gudlavalleru Engineering College
Gudlavalleru, India
dhemanth848@gmail.com

Mr. Mohammed Ezaz Ahmed
Dept.of CSE(AI&ML)
S R Gudlavalleru Engineering College
Gudlavalleru, India
ezazahmedmd555@gmail.com

Mr.Thota Abhisheka Vardhan
Dept. of CSE(AI&ML)
S R Gudlavalleru Engineering College
Gudlavalleru, India
abhishekavardhan9@gmail.com

Mr.Kodali Venkata Naga Vamsi
Dept. of CSE(AI & ML)
S R Gudlavalleru Engineering College
Gudlavalleru, India
vnvamsikodali@gmail.com

Abstract:

There are serious security dangers in cloud systems due to the quick rise in cyberthreats, especially privilege escalation assaults. Because cloud computing is centralized, it is susceptible to insider attacks, in which authorized people abuse their authority to obtain unapproved access. Insiders have authorized access, unlike external attackers, which makes it more difficult to identify their nefarious activity. This paper suggests a machine learning-based approach that combines several cutting-edge techniques to identify and prevent privilege escalation attacks. A methodical technique is created to examine user behavior and spot unusual activity that might point to security lapses. To improve model performance, the project uses hyperparameter tuning, SMOTE-based data balancing, and feature engineering. Using a stacking classifier that combines Random Forest, XGBoost, LightGBM, CatBoost, and Gradient Boosting in order to use ensemble learning.

I. Introduction

A new method for enabling and delivering services over the Internet is cloud computing. Growing computing demands and the economic crisis have altered data processing, storage, and display in the Cloud Model. Using the cloud infrastructure helps people to avoid costly equipment purchases and upkeep. Cloud storage companies protect their systems and data by means like encryption, access control, and authentication. Depending on data accessibility, speed, and frequency, the cloud may save almost any kind of data in several cloud data storage systems.

On purpose and by error, the volume of data that flows between enterprises and cloud service providers can cause sensitive data breaches. Businesses struggle to block undesired internet access due to the same features that make it easy for staff and IT systems. Due to open interfaces and authentication, cloud services put businesses at further risk for security breaches. Because of their experience, Sophisticated hackers have access to cloud systems. Using several techniques, machine learning enhances data management and security. Many datasets are private and cannot be shared because of privacy issues or absence of necessary statistical qualities. rapid expansion of the cloud sector raises legal privacy and security issues. Changes to a Cloud Company employee's function or responsibilities may not necessarily affect their access credentials. As a result, vital information is stolen and corrupted through the abuse of old rights. Every account linked to a computer has power. Usually, only approved people have access to private files, server databases, and other services. A hostile attacker can reach a sensitive system by stealing a higher user account and raising their rights. Attackers can manage the environment by moving vertically for admin and root access or horizontally to affect other systems. Horizontal privilege escalation allows a user to acquire the access rights of another user with similar access. Horizontal privilege escalation allows an attacker to obtain unneeded data. An attacker with access to poorly designed applications might find flaws in a Web application that allow him to obtain particular user data. The horizontal privilege escalation attack between the organizations' entities is depicted in Figure 1. This kind of attack often calls for a comprehensive understanding of the flaws affecting particular operating systems and the application of harmful software

II. Literature Survey

To lower false alarm rates by isolating broken and new user nodes from problematic nodes, Kumar et al. [1] suggest

https://doi.org/10.36893/FSSC.2025.V20N01.040

trend micro locality sensitive hashing (TLSH), based on clustering employing Chi-square feature selection, random forest, and PCA.

Pathy et al. claim that conventional web and cloud apps are susceptible to the most common online assaults. Pathy et al. [2]. Major SaaS application threats are SQL injection attacks. SQL attack detection classification is created and assessed using machine learning. SQL injection detection is evaluated using the AdaBoost Classifier, Random Forest, Deep Learning using ANN, TensorFlow's Linear Classifier, and Boosted Trees Classifier. Malicious writing techniques are more important than bad reading habits. The random forest classifier performs better and attains more accuracy on the dataset than any other classifier.

Sun et al. [3] claim that companies and organizations are growing increasingly dependent on the network. This creates additional network security vulnerabilities. Ponemon's 2018 Cost of a Data Breach Study finds, hostile activities accounted for 48% of data leaks from 15 countries and 17 industry groupings. However, the negligent actions of insiders were responsible for 27% of the occurrences. In this work, they employed the tree structure approach to generate the feature sequence and assess user behavior. COPOD is used to identify anomalous users and differentiate between feature patterns. Furthermore, the detecting impact performs better than the conventional unsupervised learning method. Using this approach to process large amounts of complicated and varied data has advantages.

Authorized users who steal, defraud, or disrupt private information or intellectual property are considered insider threats, according to Kim et al. [4]. Though less common than external network attacks, insider threats are nevertheless damaging. Three typical insider threat study techniques are: First, build a rule-based detection system. The second approach is to create a network graph and monitor changes to find suspect people or activity. A statistical or machine-learning model predicts hazardous conduct from past data in the third approach. They used the "CERT Insider Threat Tools" collection because it's difficult to get real business system logs. Liu and associates.

[5] pointed out that the majority of cyberattacks on information and communication technology systems come from within the company, increasing its susceptibility. Since insiders are concealed behind enterprise-level security defenses, it can be challenging to detect and minimize insider threats. By reassembling literature, they reveal the numerous kinds of insiders and their risks. Insider threats are unintentional offenders, traitors, and masqueraders. A set of defensive tactics that aid in preventing or enhancing the detection of various internal threats could be viewed as prevention. From the standpoint of data analytics, they provide the suggested initiatives in terms of contextual, network, and host data analysis.

In the interim, pertinent papers are examined and contrasted, with a brief synopsis to emphasize the advantages and disadvantages. According to [6], [7], insiders are an organization's trusted partners who have access to its networks, resources, and expertise. In 2015, insiders were accountable for almost 60% of all security breaches and assaults that were reported worldwide. Thus, preventing insider threats is a crucial concern. This paper's primary goal is to create a reliable insider threat detection system that can distinguish between malicious and benign insider activity. The primary focus of this endeavor will be the study of human behavioral activities. They examine three scenarios that are related to the insider user's behavior. The following three scenarios are:

Using a portable device to access and steal data, a user engages in post-work activities.

Some users install specific spyware programs in order to get the passwords of employees of the organization; after obtaining the passwords, they try to steal the credentials of the supervisor. Then, in an attempt to incite panic, they send the company ominous and frightening emails.

According to Tariq et al. [8], deep learning is multilevel and deep-structured machine learning algorithms that can be supervised or unsupervised. The key challenge of the DL is its learning module encrypted data and interface. Security and privacy are especially important as DL models are utilised in many different applications. Many Deep Neural Network characteristics depend mostly on input training data, hence privacy issues are always there. Deep Learning security concerns and countermeasures have been researched by academics and industry.

In their article [9], Berman et al. discussed how to safeguard the availability, confidentiality, and integrity of computer resources through a set of practices, techniques, instruments, and technology referred to as "cyber security." There is evidence of compromise at every stage of an attack. It is difficult to find these signs, which may be dispersed across the environment. Websites, apps, electronic gadgets, and other cyber-enabled products generate a lot of data. Designing generalized models for malware categorization and detection is made possible by DL. The infection is highlighted by network behavior-based methods, which are necessary for detecting complex malware.

One of the most important cyber security issues of our time is insider threat [10], [11]. Insider threats may cause a great deal of harm, yet their objectives and methods may be very different. Anomaly-based intrusion detection systems are a crucial tool for identifying known and new threats. The type of abnormality determines how to identify it. The three subtypes of anomalies are collective, contextual, and point. Intrusion based on anomalies

Detection systems model by training with "normal" network data. When completed, the system uses its model to identify unusual objects, events, and traffic. Deep

learning is a type of machine learning that uses hierarchical information processing techniques to examine or classify patterns and acquire unmonitored characteristics. The KDDCup99 system is constructed using RBM and Autoencoder. Attribute values are statistically analysed using KDD99 data. In KDDCup99 network traffic tests, Deep Learning algorithms find intrusions with low mistake rates.

Security was a major worry for Coppolino et al. [12], [13]. These hackers use their devices as a conduit to upload susceptible apps to the cloud. This code is destructive when correctly injected, and it is controlled by the person running it. The quality of the created code and the extent of the cloud's security restrictions determine how important it is that this code can give access to information by hostile users. All clouds are covered by security protections from cloud ecosystem. To safeguard privacy, user authentication data will be kept on the cloud. OTP was possible if the system's architecture took validated authentication into account.The CloudSim simulator models both the suggested system and algorithm.

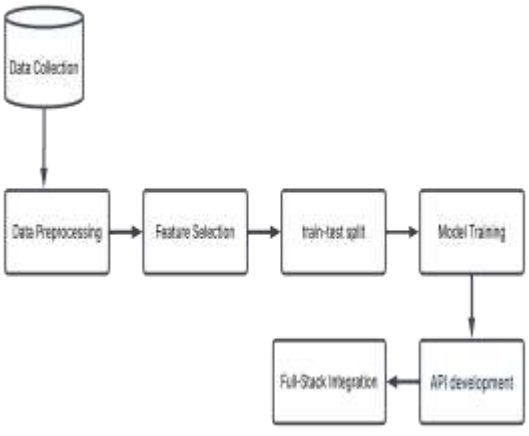
A malware detection technique (DL) based on Deep Learning was discussed by Abdelsalam et al. [14]. According to the research, raw process behaviour (performance measurements) data allows a 2D CNN to detect malware. The paper first develops a 2D CNN model without a time window and then contrasts it with a recently developed 3D CNN model that significantly increases detection accuracy by adding a time window as the third dimension, hence lowering mislabeling. 2D CNN had 79% fair accuracy on the test dataset.

Jaafar et al. [15] shown that many people are served by information systems. Therefore, different users on the same information system can have different rights. Many studies have shown security holes in privilege escalation or attacks exploiting information system anomalies and weaknesses. Without supposing any dataset or distribution assumptions, the work offers a distance-based outlier detection approach for unanticipated privilege escalation concerns. Second, the paper identifies four privilege escalation strategies and justifies their specification depending on known conditions.

III. Methodology

i) Proposed Work:

Using the Privilege Escalation Attack Detection and Mitigation system, cybersecurity threats may be categorised in real-time, a complete solution using machine learning, API development, and full-stack web apps. Each step carefully crafted to guarantee high detection accuracy, scalability, and user accessibility, the system runs a well-defined workflow shown in the flowchart. Each stage is explained more below to offer a better knowledge of the operation and execution of the system



ii) System Architecture:

Fig: 1. Proposed architecture

iii)Data Collection

Data gathering is the first phase in the process and is crucial to the system as a whole. Security logs are collected from a variety of sources, such as enterprise network access control records, intrusion detection systems (IDS), and cloud environments like AWS, Azure, and Google Cloud. Key security parameters including unsuccessful login attempts, session duration, keystroke anomalies, IP address patterns, and network violations are all included in the carefully selected dataset. In order to offer context for possible attacks, additional metadata is gathered, such as timestamps, user roles, and privilege levels. The system will have a big and diverse dataset thanks to this thorough data gathering, which captures both benign and malevolent activity. This is crucial for building a strong machine learning model that can recognize attempts at privilege escalation.

iv)**Data Preprocessing**

To get it ready for analysis, the data must go through a thorough preparation step after collection. By using imputation approaches to handle missing values—such as substituting mean or median values, depending on the feature's distribution—this stage tackles frequent problems with data quality. Z-score and Min-Max scaling are two methods used to normalize features.

standardization to avoid bias toward features with wider ranges and guarantee that all qualities contribute equally during model training. To make it compatible with machine learning algorithms, categorical data—like user roles or IP categories—is encoded using label encoding or one-hot encoding. Using the Synthetic Minority Over-sampling Technique (SMOTE), the problem of imbalanced data—where attack cases are much outnumbered by non-attack cases—is addressed. In order to ensure a balanced dataset and avoid bias in the model towards the majority class, SMOTE creates synthetic samples for the minority class (attack cases). For the model to learn efficiently and generalize well across various contexts, this preprocessing phase is essential.

v)Feature Selection

The system uses feature selection after preprocessing to determine which attributes have the greatest influence on identifying privilege escalation attacks. While Recursive

Feature Elimination (RFE) repeatedly eliminates less significant features based on their contribution to model performance, techniques like Correlation Analysis are employed to reduce highly correlated features that might generate redundancy. Furthermore, by measuring the significance of every feature in the prediction process, SHAP (SHapley Additive exPlanations) values are used to give interpretability. Unusual session durations or the quantity of unsuccessful login attempts, for instance, may be shown to be highly predictive of an attack, but less pertinent features—like timestamps with little variance—are eliminated. computational overhead, improving the system's scalability and efficiency for real-time uses.

vi)Train-Test Split

To evaluate the model's capacity to manage unrecognised data, the dataset is divided into training and testing sets. While testing utilises 20%, training uses 80% of the data. During the split, stratified sampling is used to maintain class balance by distributing attack and non-attack instances proportionately throughout the sets. By using this technique, probability of overfitting—when the model excels on training data but badly on unknown inputs—is decreased. In order to further improve the model, a small validation set—typically 10% of the training data—is also set aside during hyperparameter tweaking. This structured approach to data splitting supports a thorough evaluation, reflecting how the model might perform in real-world conditions where new, previously unseen attack patterns could appear.

vi)Model Training

To create a reliable detection system, a range of machine learning methods are used during the model training stage. The preprocessed dataset is used to train models like Random Forest, XGBoost, LightGBM, CatBoost, and Gradient Boosting, which were selected for their capacity to manage intricate, non-linear correlations in security data. To make sure these models perform effectively across several detection dimensions, a wide range of metrics are used to evaluate them, including accuracy, precision, recall, and F1-score. High recall, for example, is given priority in order to reduce false negatives and guarantee that the majority of attacks are identified, even if this occasionally results in the flagging of innocuous activity (false positives).

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	0.811667	0.533951	0.697581	0.604895
XGBoost	0.858333	0.678899	0.596774	0.635193

activities (false positives). Hyperparameter tuning is conducted using GridSearchCV, which systematically tests combinations of parameters, and TPOT AutoML, which automates the optimization process by exploring a wide range of algorithms and configurations. The final model achieves an accuracy of 87%, with a balanced F1-score that reflects its ability to detect privilege escalation attacks efficiently while maintaining a low rate of false positives. This high detection efficiency makes the system reliable for real-world deployment.

4.EXPERIMENTAL RESULTS*Fi*

4.1 MODEL EVALUATION

Accuracy:

Accuracy measures the overall proportion of correct predictions made by the model, indicating how well it performs across all classes. It is calculated as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number of Instances}}$$

While accuracy provides a general sense of performance, it may be misleading in cases of imbalanced datasets, where the number of instances in different classes varies significantly. *F1-Score:*

F1-Score is the harmonic mean of precision and recall, balancing false positives and false negatives. It is particularly useful in imbalanced datasets, ensuring a model does not favor one class over another. The F1-Score is calculated as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The F1-Score provides a single metric that balances the trade-off between precision and recall, offering a more comprehensive evaluation of model performance, especially when dealing with imbalanced classes. These metrics collectively offer a comprehensive understanding of a model's performance, each highlighting different aspects of its predictive capabilities.

LightGBM	0.858333	0.691176	0.568548	0.623894
CatBoost	0.856667	0.668142	0.608871	0.637131
Gradient Boosting	0.854167	0.669767	0.580645	0.622030

4.2 FULL-STACK INTEGRATION

The final phase is full-stack integration, where the FlaskAPI is connected to a ReactJS frontend to create a user-friendly interface for real-time attack monitoring. The frontend features an input form where users can enter security log attributes, and upon submission, the data is sent to the Flask API for prediction. The interface dynamically displays the classification result, showing "Attack Found" in red and "No Attack Found" in green. CSS enhancements, including a cybersecurity-themed background and responsive design, make the interface visually appealing. The system is designed to be deployed on cloud platforms such as AWS or Azure for live security monitoring

Fig: Result of attack not Found

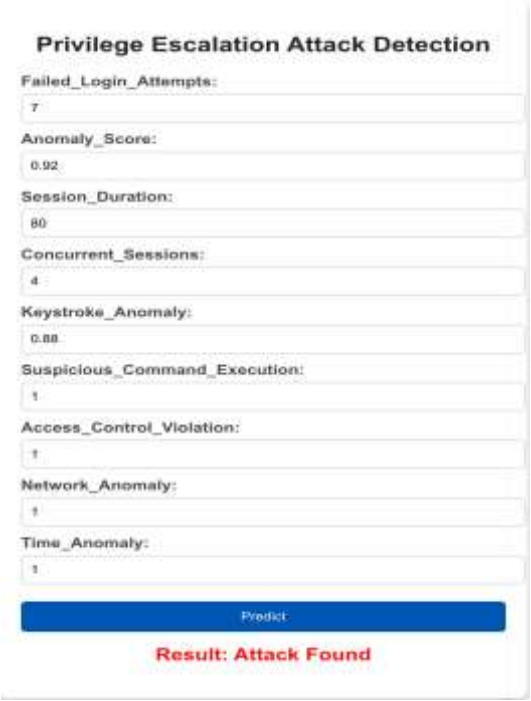


Fig : Result of attack Found

IV. CONCLUSION

The successful application of a machine learning-based privilege escalation detection system in cloud environments demonstrates its remarkable 87% accuracy rate in identifying security issues. Through the use of robust classification models, the system accurately distinguishes between benign and malicious activity, enhancing cybersecurity. Combining a Flask API with a ReactJS-based interface enables real-time monitoring, bolstering the project by enabling users to submit security-related data and receive immediate classification results. This ensures a rapid and seamless hazard detection system that is used in real-world scenarios. Hyperparameter adjustment, SMOTE-based data balance, and feature selection techniques all significantly enhance the model's predictive performance. The detection system's dependability is strengthened by these enhancements, which also decrease false positives and negatives while increasing accuracy. By combining powerful machine learning techniques with an easy-to-use user interface, our study provides a scalable and effective means of lowering privilege escalation hazards in cloud environments.

Future Scope

Incorporating temporal user behaviour research will help future work to enhance attack detection. Non-Markovian models retaining past data can improve accuracy and lower false negatives by helping to spot incremental privilege escalation attempts. Strict access control regulations, sophisticated security technologies, and monitoring of all network connections—including mobile

and remote access—can help to reinforce mitigation tactics. Automating these actions using AI-driven systems would enhance real-time threat detection and response.

Future improvements might employ deep learning methods including RNNs and transformers to identify consecutive assault patterns. Including dynamic response systems and real-time notifications will also help to improve security even more.

Wider acceptance will be guaranteed by increasing system interoperability with several cloud platforms and security frameworks. Creating an API for smooth integration with cloud service providers and security solutions would improve its usefulness in corporate settings.

.

REFERENCES

[1] R. Kumar, K. Sethi, N. Prajapati, R. R. Rout, and P. Bera, ‘‘Machine learning based malware detection in cloud environment using clustering approach,’’ in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–7.

[2] D. Tripathy, R. Gohil, and T. Halabi, ‘‘Detecting SQL injection attacks in cloud SaaS using machine learning,’’ in Proc. IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS), May 2020, pp. 145–150.

[3] X. Sun, Y. Wang, and Z. Shi, ‘‘Insider threat detection using an unsupervised learning method: COPOD,’’ in Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE), May 2021, pp. 749–754.

[4]J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, ‘‘Insider threat detection based on user behavior modeling and anomaly detection algorithms,’’ Appl.Sci., vol. 9, no. 19, p. 4018, Sep. 2019.

[5]L. Liu, O. de Vel, Q.-L. Han, J. Zhang, and Y. Xiang, ‘‘Detecting and preventing cyber insider threats: A survey,’’ IEEE Commun. Surveys Tuts., vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.

[6]P. Chattopadhyay, L. Wang, and Y.-P. Tan, ‘‘Scenario-based insider threat detection from cyber activities,’’ IEEE Trans. Computat. Social Syst., vol. 5, no. 3, pp. 660–675, Sep. 2018.

[7]G. Ravikumar and M. Govindarasu, ‘‘Anomaly detection and mitigation for wide-area damping control using machine learning,’’ IEEE Trans. Smart Grid, early access, May 18, 2020, doi: 10.1109/TSG.2020.2995313.

[8]M. I. Tariq, N. A. Memon, S. Ahmed, S. Tayyaba, M. T. Mushtaq, N. A. Mian, M. Imran, and M. W. Ashraf, ‘‘A review of deep learning security and privacy defensive techniques,’’ Mobile Inf. Syst.,vol. 2020, pp. 1–18, Apr. 2020.

[9]D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, ‘‘A survey of deep learning methods for cyber security,’’ Information, vol. 10, no. 4, p. 122, 2019.

[10]N. T. Van and T. N. Thinh, ‘‘An anomaly-based network intrusion detection system using deep learning,’’ in Proc. Int. Conf. Syst. Sci. Eng. (ICSSE), 2017, pp. 210–214.

[11] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, ‘‘Deep learning for anomaly detection: A review,’’ ACM Comput. Surv., vol. 54, no. 2, pp. 1–38, Mar. 2021.

[12] A. Arora, A. Khanna, A. Rastogi, and A. Agarwal, ‘‘Cloud security ecosystem for data security and privacy,’’ in Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng., Jan. 2017, pp. 288–292.

[13] Shaik Salma Begum et al.” GLCM of Fuzzy Clustering Means for Textural Future Extraction of Brain Tumor in Probabilistic Neural Networks”, International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-9, Issue-1, November 2019.

[14] Shaik Salma Begum et al.” Combining optimal wavelet statistical texture and Recurrent Neural Network for Tumor detection and Classification over MRI”, Multimedia Tools and Applications, ISSN 1380-7501, January 2020, Springer.

[15] Shaik Salma Begum et al. “Combining Wavelet statistical texture and recurrent neural network for tumour detection and classification over MRI”, International Journal of Engineering and Advanced Technology (IJEAT), DOI: 10.35940/ijeat.F9388.088619.

[16] Shaik Salma Begum et al.”RDNN for Classification and Prediction of Rock or Mine in Underwater Acoustics”, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8