ZKG INTERNATIONAL ISSN: 2366-1313

DETECTING PACKET LEVEL ATTACKS IN CLOUD USING MACHINE LEARNING APPROACH

[1]Mrs. B.Ratnamala,[2]Soujenya.Voggu,[3]Tadakamalla Rahul Bharani Kumar,[4]Siddamshetty Bhargav

[1,2]Assistant Professor, Dept. of CSE-Cyber Security, Geethanjali College of Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301,

bratnamala.cse@gcet.edu.in,soujenya.voggu@gmail.com

[3, 4,BTech] Student, Dept. of CSE-Cyber Security,Geethanjali College of Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301,

rahulbharani02@gmail.com,siddamshettybhargav@gmail.com

ABSTRACT:

The proliferation of smart devices has led to a surge in cyber threats, challenging traditional security measures. Cloud computing, while transformative, poses challenges for distributed security systems due to its centralized nature, increasing the risk of data breaches. Malicious insiders within organizations, with privileged access, pose a significant threat, often difficult to detect. To address this, a machine learning-based system is proposed for insider threat detection, focusing on anomalies and security issues related to privilege escalation. Ensemble learning techniques are employed to improve prediction performance. While previous studies have explored vulnerabilities in network systems, accurately identifying insider attacks remains a gap. A customized dataset derived from the CERT dataset is utilized, and four machine learning algorithms - Random Forest (RF), Adaboost, XGBoost, and LightGBM - are applied and analyzed. LightGBM emerges as the top performer with 97% accuracy, though other algorithms excel in specific attack types. Thus, combining algorithms could enhance classification across various internal threats. The study aims to develop a systematic approach to detect and classify insider threats, leveraging ensemble learning to enhance accuracy. By evaluating algorithm performance and exploring combinations, insights into effective strategies for

ISSN: 2366-1313

mitigating insider threats are provided, advancing organizational cyber security measures.

Keywords: Machine Learning,Packet Level Attacks.

I INTRODUCTION

In recent years, the proliferation of cloud computing technologies has revolutionized how organizations store, process, and manage their data. However, alongside the benefits of cloud adoption come new cyber security challenges, particularly concerning insider threats. Insider threats, perpetrated by individuals with privileged access to an organization's systems, pose significant risks to data security and integrity within cloud environments. The rise of insider threats in cloud computing necessitates robust detection and mitigation strategies to safeguard sensitive data and mitigate potential damages. Traditional cyber security measures alone may prove insufficient in addressing the dynamic and evolving nature of insider threats within cloud infrastructures. Hence, there is a growing need for advanced techniques, such as machine learning algorithms and behavioral analytics, tailored specifically for detecting and mitigating insider threats in cloud computing environments. This study aims to explore the theoretical background, methodologies, and practical applications of insider threat detection and mitigation in cloud computing. By understanding the underlying principles and challenges, organizations can develop proactive strategies to detect, respond to, and mitigate insider threats effectively. Through an interdisciplinary approach that combines cyber security principles, machine learning techniques, and cloud security practices, this study seeks to contribute to the ongoing efforts in enhancing the security posture of cloud-based systems.

II. LITERATURE SURVEY

Le et al. highlighted insider threats as one of the most expensive and challenging forms of attacks due to insiders' knowledge of the organization's systems and security processes. They emphasized the importance of machine learning in detecting insider threats, with Random Forest showing promising results in terms of detection performance and low false positive rates.

Janjua et al. focused on using machine learning approaches to classify emails in the context of insider threats. They found AdaBoost to have the best classification accuracy for distinguishing between harmful and non-malicious emails. However, they noted that the dataset used in their study was

ISSN: 2366-1313

limited, and further research with a larger dataset could improve model performance.

Kumar et al. addressed the challenges of implementing security and resilience in cloud platforms, particularly regarding malware detection. They proposed a novel malware detection technique using trend micro locality sensitive hashing (TLSH) and found that Random Forest performed the best among the classifiers tested.

Le and Zincir-Heywood discussed the difficulty in researching insider threats due to insiders' access to the organization's network systems and their familiarity with security processes. They used machine learning techniques such as Random Forest and Artificial Neural Networks (ANN) to detect harmful insider activity, achieving good results.

III SYSTEM ANALYSIS

EXISTING SYSTEM

Existing methods discussed that due to the large number of diverse apps operating on shared resources, implementing security and resilience on a Cloud platform is necessary but difficult. Inside the Cloud infrastructure. Based on the idea of clustering, a novel malware detection technique was suggested: trend micro locality sensitive hashing (TLSH). They utilized Cuckoo sandbox, which generates dynamic file analysis results by running them in a separate environment. Existing methods discussed that insider threats are among the most expensive and difficult-todetect forms of assault since insiders have access to a company's networked systems and are familiar with its structure and security processes. A unique set of challenges face insider malware detection, such as extremely unbalanced data, limited ground truth, and behavioral drifts and shifts.

Disadvantages

- Data from any provider is stored in cloud without verifying if the given packet data is secure or not which causes attacks on cloud servers.
- Data security is provided by just scanning virus for files but not the type of attack from the network.

PROPOSED SYSTEM

In order to generate findings that represent real-world situations, this work assumes a realistic context for ml model training. After this, the work emphasizes the differences from training under conventional ml conditions. Create and analyze a user-centered insider attack detection process,

ISSN: 2366-1313

including data collection, pre-processing, and ml model-based data analysis. We develop a system where user's data transfer is verified with the packet data and send to server to check status of packet as attacked packet or not based on these details decision is taken for data transferring or data storage.

Advantages

- Machine learning based algorithms are used to train network packet data to automate process of predicting if the given packet is attacked or not.
- Time taken for prediction is less and there is no need of any manual process for predicting. Detects four types of attacks and reduce changes of attacks on cloud.

IV IMPLEMENTATION

Architecture:

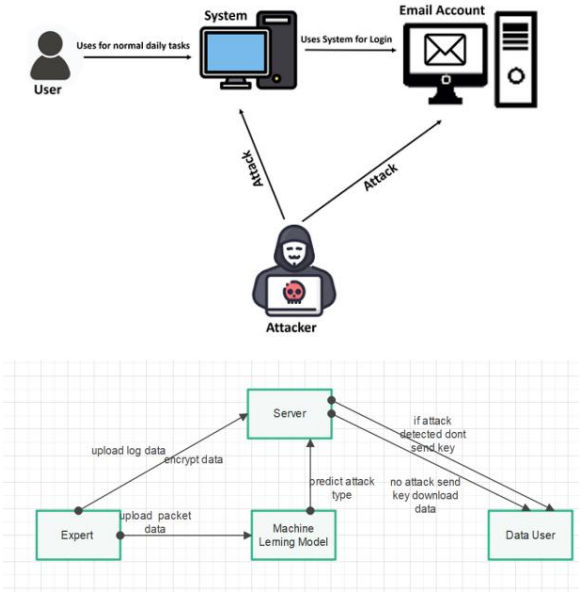


Fig-1. Architectures of the system model

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system. Organized in a way that supports reasoning about the structures and behaviors of the system.

3-Tier Architecture: The three-tier software architecture (a three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides 26 process management where business logic

ISSN: 2366-1313

and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging. The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems.

Advantages of Three-Tier:

- Separates functionality from presentation.
- Clear separation – better understanding.
- Changes limited to well define components.
- Can be running on WWW

MODULES

1. Owner module

Owner will register into the application by providing all the necessary details and therefore he can login into the application using username and password and user can upload the files to application and share with the other registered users. He can also view the files uploaded by him and can also view the requests for secret key from the other users and we can respond and the key will be sent to user by mail. Using that key, he can download the file and view the information. When owner sends data to user application will predict and check if the given packet is fraud or not using machine learning if packet is attack it is detected by machine learning and stops before upload to cloud.

2. User module

User will register with application and get username and password. Owner can see all encrypted files uploaded by all users and send request to respective user and get approval to download data and three keys for rsa are shared to owner email which can be used for owner download. Use can view data of owner if it is not attacked packet.

3. Attack detection machine learning stage

In this stage cloud network dataset is taken as input and trained using machine learning algorithms and integrated in to cloud module

ISSN: 2366-1313

where when user sends data it will check packet and predict if user is attacker or not.

4. Data collection

There are three symbolic data types in nsl-kdd data features: protocol type, flag and service. We use one-hot encoder mapping these features into binary vectors. One-hot processing: nsl-kdd dataset is processed by one-hot method to transform symbolic features into numerical features. For example, the second feature of the nsl-kdd data sample is protocol type. The protocol type has three values: tcp, udp, and icmp. One-hot method is processed into a binary code that can be recognized by a computer, where tcp is [1, 0, 0], udp is [0, 1, 0], and icmp is [0, 0, 1]

5. Pre-processing

When the dataset is extracted, part of the data contains some noisy data, duplicate values, missing values, infinity values, etc. Due to extraction errors or input errors. Therefore, we first perform data preprocessing. The main work is as follows. (1) Duplicate values: delete the sample's duplicate value, only keep one valid data. (2) Outliers: in the sample data, the sample size of missing values(not a number, nan) and infinite values(inf) is small, so we delete this. (3) Features delete and transform: in cse-cic-ids2018, we delete features such as ''timestamp'', ''destination address'', ''source address'', ''source port'', etc. If features ''init bwd win byts'' and features ''init fwd win byts'' have a value of −1, we add two check dimensions. The mark of −1 is 1. Otherwise, it is 0. In nsl-kdd, we use the one hot encoder to complete this conversion. For example, ''tcp'', ''udp'' and ''icmp'' are functions of three protocol types. After onehot encoding, they become binary vectors (1, 0, 0), (0, 1, 0), (0, 0, 1). The protocol type function can be divided into three categories, including 11 categories for flag function and 70 categories for service function. Therefore, the 41 dimensions initial feature vector becomes 122 dimensions. (4) numerical standardization: in order to eliminate the dimensional influence between indicators and accelerate the gradient descent and model convergence, the data is standardized, that is, the method of obtaining z-score, so that the average value of each feature becomes 0 and the standard deviation becomes 1, converted to a standard normal distribution, which is related to the overall sample distribution, and each sample point can have an impact on standardization. The standardization formula is as follows, u is

ISSN: 2366-1313

the mean of each feature, s is the standard deviation of each feature, and x 0 i is the element corresponding to each column's features.

6. Train-test split and model fitting

Now, we divide our dataset into training and testing data. Our objective for doing this split is to assess the performance of our model on unseen data and to determine how well our model has generalized on training data. This is followed by a model fitting which is an essential step in the model building process.

7. Model evaluation and predictions

This is the final step, in which we assess how well our model has performed on testing data using certain scoring metrics, i have used 'accuracy score' to evaluate my model. First, we create a model instance, this is followed by fitting the training data on the model using a fit method and then we will use the predict method to make predictions on x test or the testing data, these predictions will be stored in a variable called y_test_hat. For model evaluation, we will feed the y test and y_test_hat into the accuracy score function and store it in a variable called test accuracy, a variable that will hold the testing accuracy of our model.

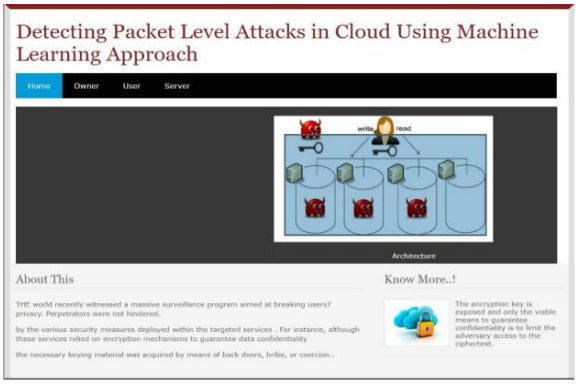
We followed these steps for a variety of classification algorithm models and obtained corresponding test accuracy scores.

ALGORITHMS

- Random Forest
- Logistic Regression (LR)
- Decision Tree (DT):
- Support Vector Machine (SVM)
- RSA Algorithm

V RESULT AND DISCUSSION

Home Page:

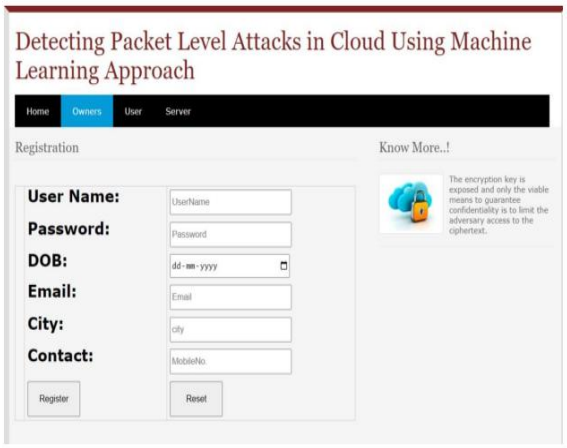


Owner Login Page:

ISSN: 2366-1313

Server Login page:

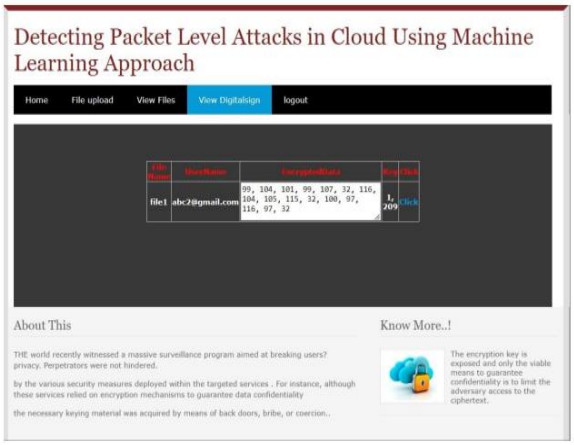
Owner Register page:



File Upload:

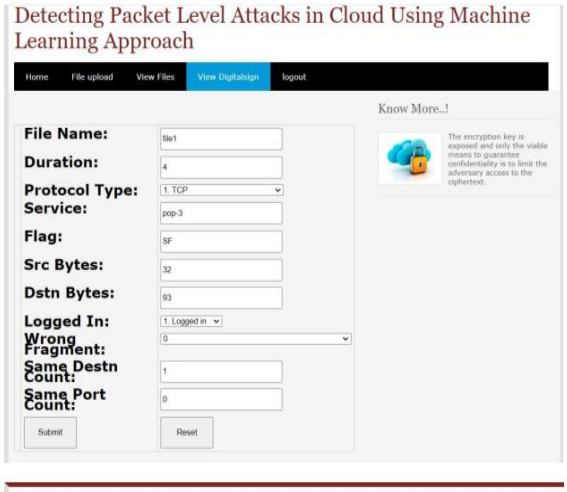
View Files:

View Digitalsign:

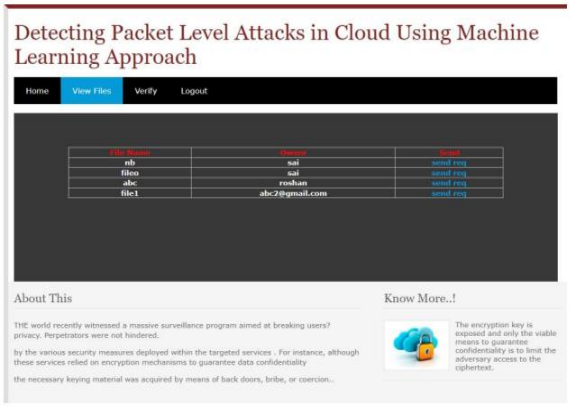


View Digitalsign:

ISSN: 2366-1313



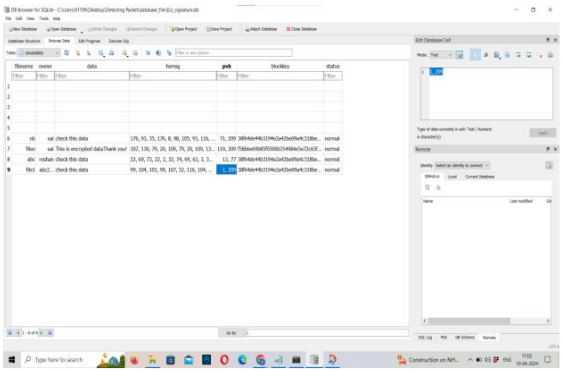
View Files:



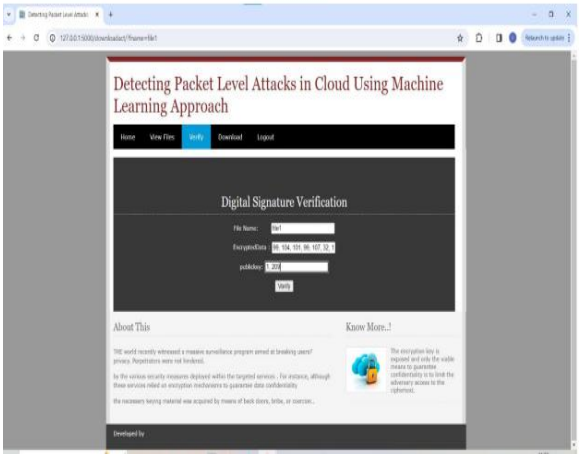
View Request:

Verify:

Verify:

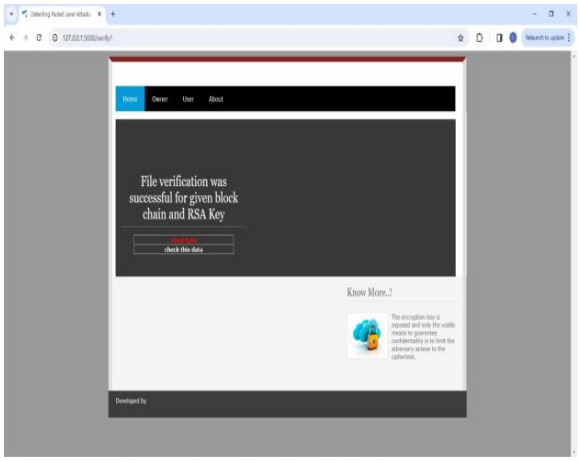


Digital sign verification:



File verification:

ISSN: 2366-1313



VI CONCLUSION

The malicious insider becomes a crucial threat to the organization since they have more access and opportunity to produce significant damage. Unlike outsiders, insiders possess privileged and proper access to information and resources. This paper proposed machine learning algorithms for detecting and classifying an insider attack. A customized dataset from multiple files of the CERT dataset is used in this work. Four machine learning algorithms were applied to that dataset and gave better results. These algorithms are Random Forest, AdaBoost, XGBoost, and LightGBM. Using these supervised machine learning algorithms, this paper demonstrated the effective experimental results having higher accuracy in the classification report. Among the proposed algorithms, the LightGBM algorithm provides the highest accuracy of 97%; the other accuracy values are RF with 86%, AdaBoost with 88%, and XGBoost with 88.27%.

FUTURE ENHANCEMENT

In the future, the proposed models may increase their performance by expanding the dataset in size and diversity in terms of its features and the new trends of insider attackers to perform the attack. This may open up new research trends toward detecting and classifying insider attacks related to many fields of organization. Machine learning models are used by businesses to make credible business decisions, and improved model results lead to better judgments. The cost of mistakes can be quite high; however, this cost is reduced by improving model accuracy. ML-based research enables users to provide massive amounts of data to computer algorithms, which then evaluate, recommend, and decide using the supplied data.

VII REFERENCES

[1] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, ‘‘Cloud-based email phishing attack using machine and deep learning algorithm,’’ Complex Intell. Syst., pp. 1– 28, Jun. 2022.

[2] D. C. Le and A. N. Zincir-Heywood, ‘‘Machine learning based insider threat modelling and detection,’’ in Proc.

IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM), Apr. 2019, pp. 1–6.

[3] P. Oberoi, ‘‘Survey of various security attacks in clouds based environments,’’ Int. J. Adv. Res. Comput. Sci., vol. 8, no. 9, pp. 405–410, Sep. 2017.

[4] A. Ajmal, S. Ibrar, and R. Amin, ‘‘Cloud computing platform: Performance analysis of prominent cryptographic algorithms,’’ Concurrency Comput., Pract. Exper., vol. 34, no. 15, p. e6938, Jul. 2022.

[5] U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, ‘‘Cloud security threats and solutions: A survey,’’ Wireless Pers. Commun., vol. 128, no. 1, pp. 387–413, Jan. 2023.

[6] H. Touqeer, S. Zaman, R. Amin, M. Hussain, F. Al-Turjman, and M. Bilal, ‘‘Smart home security: Challenges, issues and solutions at different IoT layers,’’ J. Supercomput., vol. 77, no. 12, pp. 14053–14089, Dec. 2021.

[7] S. Zou, H. Sun, G. Xu, and R. Quan, ‘‘Ensemble strategy for insider threat detection from user activity logs,’’ Comput., Mater. Continua, vol. 65, no. 2, pp. 1321–1334, 2020.

[8] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, ‘‘On the effectiveness of machine and deep learning for cyber security,’’ in Proc. 10th Int. Conf. Cyber Conflict (CyCon), May 2018, pp. 371–390.

[9] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, ‘‘Analyzing data granularity levels for insider threat detection using machine learning,’’ IEEE Trans. Netw. Service Manag., vol. 17, no. 1, pp. 30–44, Mar. 2020.

[10] F. Janjua, A. Masood, H. Abbas, and I. Rashid, ‘‘Handling insider threat through supervised machine learning techniques,’’ Proc. Comput. Sci., vol. 177, pp. 64–71, Jan. 2020.

[11] Prasadu Peddi (2015) "A review of the academic achievement of students utilising large-scale data analysis", ISSN: 2057-5688, Vol 7, Issue 1, pp: 28-35.

[12] 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.

[13] Prasadu Peddi (2015) "A machine learning method intended to predict a student's academic achievement", ISSN: 2366-1313, Vol 1, issue 2, pp: 23-37.

AUTHORS

Mrs. B.Ratnamala, Assistant Professor Dept. of CSE-Cyber Security,Geethanjali College of

ISSN: 2366-1313

Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301.

Email:bratnamala.cse@gcet.edu.in

Mrs. Soujenya.Voggu, Assistant Professor Dept. of CSE-Cyber Security,Geethanjali College of Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301.

Email: soujenya.voggu@gmail.com

Mr.Tadakamalla Rahul Bharani Kumar, Dept. of CSE-Cyber Security, Geethanjali College of Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301.

Email: rahulbharani02@gmail.com

Mr. Siddamshetty Bhargav, Dept. of CSE-Cyber Security,Geethanjali College of Engineering and Technology Cheeryal (V), Keesara (M), Medchal(D), Hyderabad, Telangana 501301.

Email: siddamshettybhargav@gmail.com