Int.J.HIT.TRANSC:ECCN.Vol.10: Special Issue (2025) Page 120-124 ISSN: 0973-6875

Available Online at www.hithaldia.in/locate/ECCN

ORIGINAL CONTRIBUTION

Host Based Intrusion Detection System

[1]Mehaboob Mujawar, [2]Aasheesh Raizada

[1]*Research Scholar, Mangalayatan University, Beswan, Aligarh-India*

[2,]*Associate Professor, Mangalayatan University, Beswan, Aligarh-India*

ABSTRACT

This paper develops a Host-Based Intrusion Detection System (HIDS) using the Isolation Forest algorithm to detect anomalies in host system logs, such as CPU usage and memory activity. The system employs a user-friendly GUI for data input, anomaly visualization, and result analysis. Optimized for real-time detection, it addresses challenges like high false positive rates and scalability. By leveraging machine learning, this research enhances the detection of both known and novel threats, contributing to robust cybersecurity solutions and paving the way for future integrations with network-based intrusion detection systems.

KEYWORDS: *Host-Based Intrusion Detection System (HIDS), Anomaly Detection, Isolation Forest Algorithm, System Logs, Cybersecurity, Machine Learning, Real-Time Threat Detection, Resource Usage Monitoring, False Positive Reduction, Graphical User Interface (GUI)*

1. INTRODUCTION

Intrusion Detection Systems (IDS) are integral to modern cybersecurity frameworks, designed to safeguard computer systems and networks from unauthorized access, malicious activities, and potential security breaches. IDS solutions are broadly categorized into Network-Based Intrusion Detection Systems (NIDS) and Host-Based Intrusion Detection Systems (HIDS). While NIDS monitor network traffic for suspicious activity, HIDS focuses on monitoring individual host systems by analyzing system logs, resource usage, file integrity, and user activities. The growing complexity and volume of cyber threats have underscored the importance of HIDS in identifying internal security threats, such as those stemming from malicious insiders or unauthorized changes. Traditional IDS primarily rely on signature-based detection methods, which identify threats based on predefined patterns of known attacks. While effective against previously encountered threats, these systems struggle to detect novel or zero-day attacks that lack existing signatures. This limitation highlights the need for anomaly detection techniques, which identify deviations from normal behavior to flag potential security incidents.

Machine learning algorithms, such as Isolation Forest, offer a promising approach to anomaly detection by learning normal system behavior and identifying anomalies as deviations. These techniques are particularly valuable in analyzing high-dimensional datasets, such as system logs, where traditional methods may falter. By leveraging machine learning, HIDS can enhance the early detection of threats and reduce response times, strengthening overall security posture. This project aims to develop a Host-Based Intrusion Detection System employing Isolation Forest to detect anomalies in system logs. The system analyzes metrics such as CPU usage, memory consumption, and disk activity to identify abnormal patterns. A user-friendly Graphical User Interface (GUI) has been integrated, enabling users to upload logs, visualize detection results, and save outputs for further analysis. Through this research, the project seeks to address key challenges in HIDS, such as real-time

*Corresponding Address: mehaboob311134@gmail.com

DOI: 10.5281/zenodo.14592704

detection, scalability, and reducing false positive rates, thereby advancing the state-of-the-art in host-level cybersecurity solutions.

2. LITERATURE SURVEY

In the field of intrusion detection, several studies have explored different methodologies for identifying threats and anomalies in both network and host-based systems. Traditional intrusion detection systems relied on signature-based detection, where attacks were identified based on predefined patterns or signatures of known threats. While effective against known attacks, this approach struggles to identify novel or zero-day threats that do not match any existing signature. In response, the need for anomaly detection systems became apparent, as they offer a way to detect previously unknown attacks by identifying deviations from normal behavior. A variety of machine learning techniques have been applied to anomaly detection in IDS systems, including K-means clustering, Support Vector Machines (SVMs), Auto encoders, and Isolation Forest. Among these, Isolation Forest has gained significant attention due to its efficiency in high-dimensional spaces and its ability to detect anomalies in large datasets. Isolation Forest works by isolating observations using decision trees, where the anomalies are those that can be isolated with fewer splits compared to normal data points. This makes it highly effective for anomaly detection, particularly in scenarios where the data is sparse or high-dimensional, as is often the case with system logs.

HIDS, when combined with machine learning techniques like Isolation Forest, offers a powerful way to detect internal threats or issues that may not be easily detectable through traditional monitoring methods. Studies have demonstrated the effectiveness of applying machine learning algorithms to HIDS, particularly for tasks such as detecting abnormal system behavior, unauthorized access, and malicious software activity. However, implementing machine learning in HIDS also comes with challenges, such as feature selection, data preprocessing, and managing false positive rates. Additionally, most existing research has focused on offline analysis, where logs are processed after the fact. Real-time detection, which is crucial for preventing immediate threats, remains an area that requires further development. One of the key issues highlighted in the literature is the trade-off between detection accuracy and false positive rates.

While machine learning algorithms can detect a wide range of anomalies, they are often prone to generating false positives, which can overwhelm system administrators and reduce the effectiveness of the IDS. Another challenge is scalability, as many machine learning-based IDS systems struggle to handle the volume of data generated by large organizations with multiple host systems. Thus, while significant progress has been made in applying machine learning to intrusion detection, there are still several gaps in real-time processing, scalability, and reducing false positives. The literature review highlights the significant progress made in the application of anomaly detection for Host-Based Intrusion Detection Systems (HIDS), particularly with the use of machine learning techniques such as Isolation Forest. While this approach offers substantial advantages, including the ability to detect unknown threats, several challenges remain, particularly in the areas of real-time detection, scalability, and reducing false positives. Future research in this area will likely focus on improving the efficiency of anomaly detection algorithms, exploring hybrid models that combine multiple techniques, and enhancing real-time detection capabilities. Integrating HIDS with other cybersecurity measures, such as network intrusion detection systems (NIDS), could also provide a more comprehensive defense against sophisticated cyber threats

3. RESEARCH METHODOLOGY

The research methodology for this project adopts a systematic approach to developing and evaluating a Host-Based Intrusion Detection System (HIDS) using anomaly detection techniques. The process begins with defining the research problem, focusing on improving HIDS capabilities to detect both known and novel intrusions. System logs containing metrics such as CPU usage, memory consumption, disk

activity, and event types were collected and preprocessed through cleaning, normalization, and encoding to ensure compatibility with the Isolation Forest algorithm. Feature selection and engineering were conducted to extract relevant attributes that enhance the model's detection accuracy. The Isolation Forest algorithm was implemented in Python, utilizing libraries like Scikit-learn for efficient anomaly detection in high-dimensional datasets. A user-friendly Graphical User Interface (GUI) was developed using Tkinter, enabling users to upload logs, visualize results through charts, and save outputs for further analysis. The system was evaluated using key performance metrics such as precision, recall, F1-score, and Area Under the Curve (AUC) to ensure its effectiveness. Optimization involved fine-tuning hyper parameters to balance accuracy and false positive rates, alongside iterative testing for scalability and real-time.

The Graphical User Interface (GUI) of the system is designed to provide a seamless user experience for interacting with the anomaly detection tool. The Load Dataset button allows users to upload the input CSV dataset containing system logs. Once the dataset is loaded, the Run Anomaly Detection button initiates the detection process, applying the Isolation Forest algorithm to identify anomalies. After the detection process is complete, users can click the Visualize Results button to generate visual outputs such as charts and graphs, which help in analyzing the identified anomalies and overall system behavior. Finally, the Save Results button enables users to save the processed data into output files, categorizing records into anomalies and normal data for further investigation or reporting. This intuitive interface ensures accessibility and efficiency, making the system usable for technical and non-technical users alike. integration. Finally, the system was tested with simulated live data to validate its real-time anomaly detection capabilities. This methodology ensures the development of a robust, scalable, and efficient HIDS that addresses critical cybersecurity challenges.

Figure 1: Graphical user interface

The scatter plot for CPU usage visually depicts the usage patterns of different records within the dataset. Each point on the graph corresponds to an individual record, where normal records are displayed in green, and anomalous records, such as those indicating high CPU usage, are highlighted in red. This visual representation helps identify anomalies by showcasing instances where CPU usage exceeds predefined thresholds or deviates significantly from typical behavior.

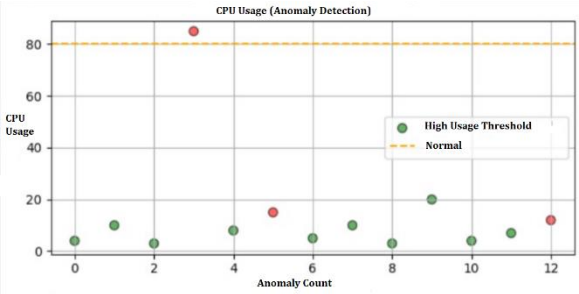

Figure 2: Scatter Plot for CPU Usage with Anomalies

By providing an intuitive and clear visualization, this graph aids in quickly pinpointing irregularities in system performance for further analysis.

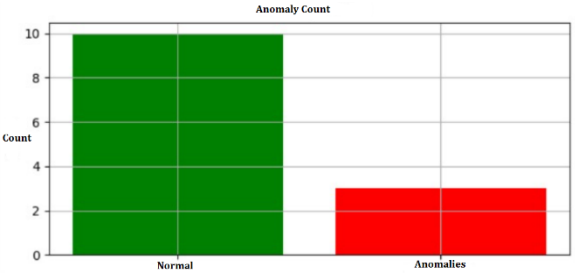

Figure 3: Bar Plot for Anomaly Counts

The bar plot for anomaly counts provides a clear summary of the detected records, categorized into "Normal" and "Anomalies." Each bar is labeled to represent the respective category, offering a quick and quantitative analysis of the dataset. This visualization highlights the balance between normal and suspicious activities, enabling users to assess the overall detection results effectively and gain insights into the distribution of anomalies within the dataset.

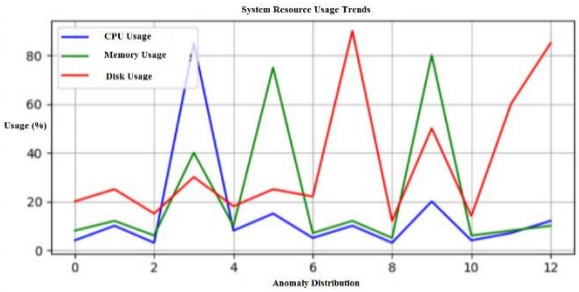

Figure 4: Line Plot for System Resource Trends

The line plot for system resource trends illustrates the patterns of CPU usage, memory usage, and disk usage over time or across record indices. Multiple lines in distinct colors represent each resource, allowing for easy comparison of their usage trends. This visualization helps identify spikes, unusual changes, or consistent patterns in resource consumption, providing valuable insights into system performance. By highlighting these trends, the line plot enables users to monitor resource usage effectively and detect potential anomalies related to system behavior.

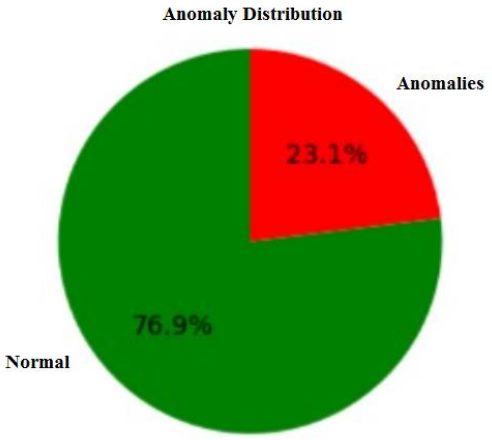

Figure 5: Pie Chart for Anomaly Distribution

The pie chart for anomaly distribution provides a clear and visually appealing representation of the dataset's composition, highlighting the proportion of normal versus anomalous records. The chart is divided into two distinct segments, with green representing normal records and red indicating anomalous records. Percentage labels are prominently displayed, making it easy to interpret the ratio of anomalies detected within the dataset. This chart serves as an intuitive and effective tool for quickly understanding the anomaly detection results.

4. CONCLUSION AND FUTURE SCOPE

In conclusion, this project successfully developed a Host-Based Intrusion Detection System (HIDS) employing the Isolation Forest algorithm for anomaly detection. The system demonstrates the ability to effectively identify deviations from normal behavior in system logs, offering a reliable approach to detecting potential security breaches. The graphical user interface enhances usability, enabling non-technical users to analyze logs and detect anomalies seamlessly. While the results show promise, challenges such as reducing false positives and optimizing real-time performance persist. Future work could focus on integrating advanced machine learning models, such as deep learning or hybrid approaches, to enhance detection accuracy and scalability. Real-time detection capabilities could be further refined by leveraging distributed computing and cloud-based systems. Additionally, exploring privacy-preserving techniques and explainable AI can make the system more transparent and secure, addressing concerns related to data sensitivity and trustworthiness. These advancements could contribute significantly to creating a robust, adaptive, and comprehensive cybersecurity framework.

REFERENCES

[1] Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (IDPS) (NIST Special Publication 800-94). National Institute of Standards and Technology

[2] Soni, S., & Tiwari, S. (2015). A survey on host-based intrusion detection system. International Journal of Computer Applications, 118(24), 5–9.

[3] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Proceedings of the IEEE Symposium on Security and Privacy, 305–316

[4] Babus, S., & Kaur, G. (2014). A review of intrusion detection systems. International Journal of Computer Applications, 88(9), 15–18.

[5] Alsulami, A. A., & Alturki, B. (2025). Enhancing multiclass network intrusion detection systems using continuous wavelet transform on network traffic. Data and Metadata, 4, 474.

[6] Alzu'bi, A., Darwish, O., Albashayreh, A., & Tashtoush, Y. (2024). Cyberattack event logs classification using deep learning with semantic feature analysis. Computers & Security, 104, 104222.

[7] Mutambik, I. (2024). An efficient flow-based anomaly detection system for enhanced security in IoT networks. Sensors, 24(22), 7408.

[8] Saranya, N., & Haldorai, A. (2024). Efficient intrusion detection system data preprocessing using deep sparse autoencoder with differential evolution. IET Information Security, 2024(1).

[9] Alfriehat, N., Anbar, M., Aladaileh, M., Hasbullah, I., Shurbaji, T. A., Karuppayah, S., & Almomani, A. (2024). RPL-based attack detection approaches in IoT networks: Review and taxonomy. Artificial Intelligence Review, 57(9).

[10] Zohourian, A., Dadkhah, S., Molyneaux, H., Pinto Neto, E. C., & Ghorbani, A. A. (2024). IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks. Computers & Security, 104, 104034.

[11] Lin, Y.-D., Yang, S.-Y., Sudyana, D., Yudha, F., Lai, Y.-C., & Hwang, R.-H. (2024). Two-stage multi-datasource machine learning for attack technique and lifecycle detection. Computers & Security, 104, 103859.

[12] Vargheese, M., Nallasivan, G., Ponkumar, D. D. N., Ponnithish, N., Karunya Devi, P., & Arun, M. (2023). Machine learning for enhanced cybersecurity. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 709–713).