Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

CloudGuard:Cloud Storage Misconfiguration and Data Leakage Monitoring System

Mynumpati Sai Prajwal
Student, BE Information Technology
MVSR Engineering College
Nadergul, Hyderabad

Keerti Kolla
Student, BE Information Technology
MVSR Engineering College
Nadergul, Hyderabad

Kadire Rishitha Reddy
Student, BE Information Technology
MVSR Engineering College
Nadergul, Hyderabad

Ch. Samson
Professor & AHOD
Department of Information Technology
MVSR Engineering College
Nadergul, Hyderabad

Abstract **- With increasing reliance on cloud computing platforms, data breach incidents due to configuration mistakes and malicious attacks are rampant. In this paper, we present CloudGuard, which is an automated cloud security monitoring tool that identifies anomalous actions by the users. This system uses the Flask backend combined with Firebase Firestore, providing constant monitoring and synchronization with an interactive dashboard in nearly real-time. It uses rule-based and Isolation Forest-based algorithms for identifying anomalous behaviors, including data leakage, login failure, and logins from unknown IPs. We have simulated the creation of cloud events for testing purposes. The visualization component uses dashboards, graphs, and geographic mapping for better security monitoring and analysis.**

Keywords - Cloud Security, Data Leakage Detection, Cloud Misconfiguration, Anomaly Detection, Isolation Forest, RealTime Monitoring, Firebase Firestore, Flask, Intrusion Detection System, Security Dashboard

I. INTRODUCTION

Due to recent advancements in technology, cloud computing has become very important for providing different kinds of services for storing and managing information. In contrast, there are certain threats associated with cloud computing which can lead to leaking of information stored on cloud computing systems, which can harm the privacy and confidentiality of the data.

Traditional security strategies are based on using static rules, which cannot be used for addressing the problems faced by cloud computing environments. To address this issue, it is necessary to implement an approach that monitors the activities of users on the basis of their behavior, which can include accessing cloud computing resources using multiple IP addresses frequently.

In order to counter the above-discussed threat in cloud computing, we developed a security monitoring system known as CloudGuard which makes use of Flask backend and Firebase Firestore for refreshing the data. This system also consists of a risk analysis technique based on rules. Furthermore, this proposed solution consists of isolation forest modeling for anomaly detection. A simulated environment has been used in order to produce real-time cloud environment data.

The rest of the paper is organized as follows. Section II presents the related work. Section III describes the proposed method. Section IV deals with the implementation, Section V presents results and discussion and Section VI draws conclusions.

II. RELATED WORK

There have been several scholarly pieces published that discuss cloud security issues like data breaches caused by misconfigurations and unauthorized access. For instance, Subashini and Kavitha explored different security issues related to cloud computing, such as the threats posed by cloud computing models and access control violations. Although the study stresses the need for developing secure systems, it only discusses security issues related to cloud computing.

The Isolation Forest framework was introduced by Liu et al. It is a technique used in anomaly detection. It is widely adopted nowadays due to its effectiveness in detecting anomalies in large datasets. This method can be used to detect new attacks by isolating anomalies in the data. However, visualization and real-time detection were not mentioned in the paper.

Industry-based platforms like AWS and Google Cloud have provided various security measures for ensuring that there is no data misconfiguration and also provide monitoring capabilities for detecting the vulnerability process. Even though industry-based platforms are efficient enough to tackle issues in security, they are hard to implement and lack the feature of live data visualization. The existing works also do not integrate rule-based risk analysis with anomaly

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

detection or provide simulation-based testing for generating realistic cloud events.

III. PROPOSED METHOD

CloudGuard is an online monitoring tool used in detecting cases of data breach and unusual user behavior. CloudGuard has been designed by using the front end interface, Flask backend system, and Firebase Firestore database for real-time data analysis. The backend APIs are used for handling data processing and communication between components.The logs of users including login activity, file activity, and IP addresses have been analyzed by the system. A simulation module is used to generate realistic cloud activity data for testing and demonstration purposes.

In the backend system, data analysis has been carried out by the Isolation Forest approach. The system also incorporates a rule-based risk analysis mechanism to enhance detection accuracy. The results of data analysis have been stored in Firebase and presented live in the front-end application. A risk score is calculated based on the analysis of user activity. Alerts and visual graphs can be shown to the users on the front end.Fig.1 shows the interaction between frontend, Flask backend, and Firebase. User activity is analyzed using rule-based logic and Isolation Forest in the backend. The results are stored in Firebase and displayed on the dashboard in real time.

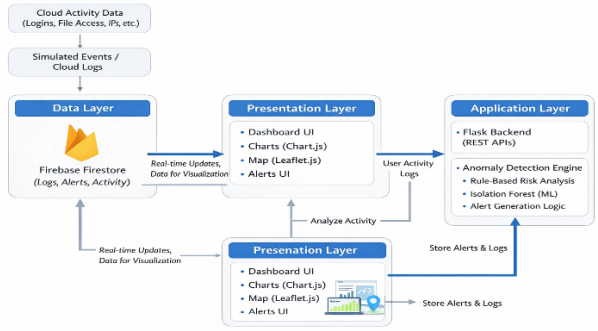

Fig.1 CloudGuard Architeture

A. Da abase Integration

It is worth mentioning that Firebase Firestore is employed as a real-time database for storing and managing all information related to user activity as well as any detected alerts. In this regard, there is smooth data synchronization between the backend part and the frontend dashboard. The former is responsible for writing processed information to Firestore, whereas the latter reads data from Firestore by means of real-time listeners.

B. R al-Time Threat Monitoring

CloudGuard constantly tracks user activities in the cloud computing infrastructure for any possible threat that may arise. It analyzes the data collected from various activities including logins, files accessed, and IP addresses in realtime. Consequently, any unusual activity becomes known instantaneously, and its results become available on the dashboard automatically without necessitating manual updates.

C. Anomaly-Based Threat Detection

The isolation forest algorithm is used by this system to detect any abnormalities in the users' behavior. Rather than depending only on the pre-set rules to detect any attack, it looks at the anomalies in behavior to find any possible attack. In this way, it becomes easier for the system to detect any previously unknown attacks such as accessing strange IP addresses.

D. eal-Time Alerts and Severity Classification

The threats that have been identified are then classified according to the extent of the impact. They include categories like Critical, High, and Medium depending on their respective risks. Alerts for these threats will be generated immediately with prominent placement on the dashboard interface.

E. Geo ocation-Based Attack Visualization

The solution includes geolocation maps that show where the suspicious behavior originated. By tracking the IP address and plotting the information on a map, users will see graphically the location of possible attackers. The use of such functionality increases situational awareness and highlights any abnormalities regarding geographic areas.

IV. IMP EMENTATION

A. ontend Development

CloudGuard frontend is created by means of HTML, CSS, and JavaScript to ensure user-friendliness. The system is implemented using React to enhance modularity and performance. In this regard, the dashboard is created in the way that it reflects Security Operations Center (SOC). Graphical representation using libraries like Chart.js ensures presentation of graphs, bars, and pie charts. Meanwhile, leaflet.js ensures geo-mapping of suspicious IP addresses. Navigation between various elements of the software is easy since they are all created under one interface. This ensures smooth interaction among the elements of the interface for easy identification and visualization of data. In this regard, real-time updates are incorporated into the interface for seamless monitoring.

B. ackend Development

Python Flask framework is used for the implementation of the backend, and it serves as the backbone processor of the system. This framework is capable of handling user input data in terms of logs, analyzing it, and conducting an anomaly detection process. A simulation module is used to generate realistic cloud activity data for testing and evaluation.The role of the backend is to detect anomalies on the basis of some specified criteria and machine learning outcomes. Moreover, communication between frontend and backend is achieved through backend APIs for data exchange. The choice of Flask is because of its light weight and ease of integration.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

C. M hine Learning Integration

This system uses an algorithm for anomaly detection in the form of the Isolation Forest method. The system also incorporates a rule-based risk analysis mechanism to enhance detection accuracy. It serves an important function of spotting any abnormal user activity through the process of training the machine learning model on data regarding normal activity and detecting anomalous data points through the separation of abnormal from normal. Some of the potential threats that this system can detect include data exfiltration, brute force attacks, and login attempts from unauthorized IP addresses.

D. Database and Real-Time Synchronization

Firebase Firestore will be used as the real-time database where the users' logs and alerts are stored. While the backend uploads the processed data into the Firestore database, the frontend pulls the changes through the realtime listener feature. As such, any changes made to the data will automatically show up on the dashboard without the need for refreshing.

E. Alert and Visualization System

The system has a well-thought-out alerting and visualizing system to help users be more aware and respond quickly. The threats are classified according to their level of severity, such as critical, high, and medium, and are presented in the alert tab. There is also the use of visual tools like graphs and maps that show clearly where the threats are located. The system can also be used to notify users through email alerts.

V. ESULTS AND DISCUSSION

The solution, referred to as CloudGuard, was evaluated based on cloud operations, which were simulated and included both standard procedures and potential security vulnerabilities. The anomaly detection algorithm referred to as Isolation Forest successfully detected anomalies, such as frequent log-ins, abnormal operations, and login attempts from unknown IP addresses. The solution also includes a rule-based approach to risk analysis, which can help increase the precision of detection. Events could be prioritized depending on their importance, and, therefore, assist in determining whether the event under consideration is vital or not. By incorporating Firebase Firestore into the solution, it was possible to receive real-time updates and visualize all events instantly on the dashboard.Fig.2 shows dashboard that displays real-time alerts, user activity, and anomaly detection results using charts and maps. It helps in identifying suspicious behavior quickly. Data updates automatically through Firebase without manual refresh..

Security events could be visualized in real time using an interactive dashboard. The process involved the use of charts, alerts table, and the geographical location of any malicious activity. This solution made it easier to monitor suspicious activity all the time. The prioritization of events according to their urgency increased the effectiveness of dealing with security breaches.

In conclusion, it is fair to state that the system worked quite well in identifying potential scenarios when there could be data leakage. It should be noted that machine learning along with rule-based approach proved to be highly effective. Nonetheless, although tests were carried out on simulated data, real data usage can improve the performance of the system significantly.

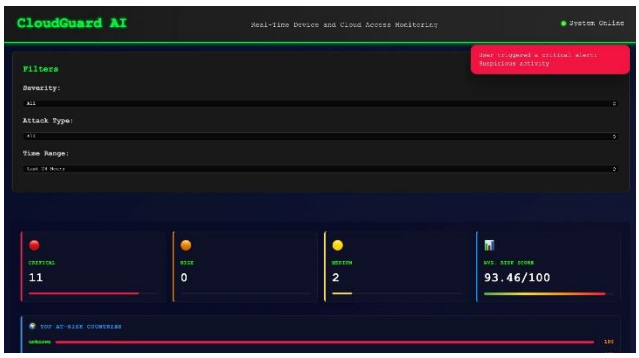

Fig.2 CloudGuard Dashboard

VI. CONCLUSION

In this study, the cloud security monitoring framework called CloudGuard has been designed. It has been designed with the aim of monitoring any leakage of information and the activity of the users in the cloud environment. An anomaly detection framework, with the application of the Isolation Forest algorithm and rule-based risk analysis framework, together with the real-time synchronization of the data using Firebase Firestore can lead to instant action being taken on the threats. Based on our research findings, the system proposed in this study is effective and feasible and can be used for the purposes as stated above. Possible areas of future study could be the use of real data and Machine Learning techniques.

The system can be improved further by integrating the system with actual cloud computing providers such as AWS, Microsoft Azure, or Google Cloud Computing, among others.

REFERENCES

[1] Gartner, "Is the Cloud Secure? Understanding Cloud Misconfigurations," Gartner Research, 2020.

[2] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[4] Amazon Web Services, "AWS Security Best Practices," 2022. [Online]. Available: https://aws.amazon.com

[5] OWASP Foundation, "Top 10 Cloud Security Risks," 2021. [Online]. Available: https://owasp.org

[6] National Institute of Standards and Technology (NIST), "Security and Privacy Controls for Information Systems and Organizations," NIST SP 800-53, 2020.

[7] Google, "Cloud Firestore Documentation," Firebase, 2023. [Online]. Available: https://firebase.google.com