IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

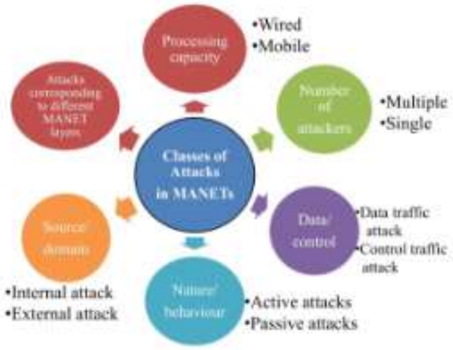
Detecting And Notifying Users Of Suspicious Activities In Real Time

[1]Dr. Palson kennedy, [2]Harini K, [3]Jayashree V
[1] Professor , [2,3]Student,
Department of Computer Science and Engineering,
PERI Institute of Technology Chennai, India

ABSTRACT: Detecting and notifying users of suspicious activities in real-time is imperative for maintaining security in various domains. This abstract presents a robust framework for achieving this goal, leveraging advanced algorithms and real-time monitoring techniques. By continuously analyzing user behavior and system interactions, our system can swiftly identify anomalies indicative of potential security breaches or unauthorized access attempts. Once detected, immediate notifications are triggered,alerting relevant stakeholders and enabling prompt response measures. Through the integration of machine learning modelsand anomaly detection algorithms, our solution ensures adaptive and proactive threat mitigation, enhancing overall system security and user trust. Detecting and notifying users of suspicious activities in real-time is a critical aspect of modern security frameworks, particularly in digital environments where threats can evolve rapidly. Our framework builds upon cutting-edge technologies, seamlessly integrating advanced algorithms with real-time monitoring capabilities to provide comprehensive protection. By continuously scrutinizing user behavior and system interactions, our system swiftly identifies deviations from established norms, flagging potential security breaches or unauthorized access attempts. These anomalies trigger immediate notifications, alerting relevant stakeholders and facilitating swift response actions. Through the fusion of machine learning models and anomaly detection algorithms, our solution not only adapts to emerging threats but also proactively anticipates them, bolstering system resilience and fostering user confidence in an ever-evolving security landscape. Safe-guarding digital ecosystems against a myriad of threats, ranging from cyber-attacks to insider threats. Our innovative framework leverages a multifaceted approach, seamlessly weaving together state-of-the-art algorithms and dynamic monitoring mechanisms to fortify defenses. By meticulously analyzing user interactions and system behavior in real-time, our system swiftly discerns deviations from normal patterns, swiftly flagging potential security breaches or illicit activities. These alerts are not only triggered instantaneously but are also accompanied by contextual insights, empowering stakeholders to make informed decisions and execute timely response strategies.

I INTRODUCTION

In an increasingly interconnected digital landscape, the detection and timely notification of suspicious activities are paramount for ensuring the integrity and security of online systems. With cyber threats evolving in sophistication and frequency, proactive measures are essential to safeguarding sensitive data and maintaining user trust. In this context, the development of advanced frameworks capable of detecting anomalies in real-time and promptly alerting users to



potential security breaches becomes indispensable. Leveraging cutting- edge technologies such as machine learning and real-time monitoring, these frameworks enable swift identification of aberrant behavior and facilitate immediate response actions.

This paper explores the significance of detecting and notifying users of suspicious activities in real-time, highlighting the pivotal role played by innovative solutions like ChatGPT in enhancing security measures and fortifying digital defenses against emerging threats. In today’s interconnected digital ecosystem, the detection and timely notification of suspicious activities are critical components of robust cyber security strategies. As cyber threats continue to proliferate and evolve in complexity, organizations and individuals alike face heightened risks of data breaches, fraud, and unauthorized access. Addressing these challenges requires proactive measures to identify and mitigate potential security vulnerabilities in real-time. This paper delves into the significance of detecting and promptly notifying users of suspicious activities, emphasizing the pivotal role played by innovative solutions like Chat GPT in bolstering cyber security defenses.

By leveraging state-of-the-art technologies such as machine learning, anomaly detection, and real-time monitoring, modern frameworks empower organizations to monitor user behavior and system interactions continuously. This proactive approach enables the swift identification of anomalies that may indicate malicious intent or unauthorized access attempts. Moreover, by integrating contextual insights and historical data, these frameworks can discern patterns indicative of emerging threats, allowing for preemptive action to be taken. The ability to detect and notify users of suspicious activities in real-time serves as a crucial line of defense against a wide array of cyber threats, including phishing attacks, malware infections, and insider threats.

II. PROPOSED FRAMEWORK

Our proposed system aims to develop a secure chat application using modern web technologies such as Socket.io, Express.js, HTML, CSS, and JavaScript. The application will provide users with a platform for real-time communication while ensuring the privacy and security of their messages. The application will support real-time messaging between users, allowing them to exchange text messages instantly. To ensure the confidentiality of messages, the system will implement end-to-end encryption using robust encryption algorithms. This will prevent unauthorized access to message content, even if intercepted during transit. The system will allow users to register their accounts securely and manage their profiles. Users can update their information, change passwords, and customize their profiles according to their preferences. The application will feature a responsive user interface designed using HTML and CSS to ensure compatibility across different devices and screen sizes. This will provide users with a seamless experience whether they access the application from a desktop or mobile device. The application will be deployed using a cloud-based infrastructure to ensure availability and reliability. Continuous monitoring and maintenance will be performed to address any issues and ensure optimal performance. Overall, our proposed chat application will offer users a secure and efficient platform for real-time communication, with features designed to enhance usability, privacy, and scalability. Through the use of modern web technologies and best practices in software development, we aim to deliver a robust and reliable solution that meets the needs of our users. The system will gather data from multiple sources including network traffic, system logs, user activity logs, endpoint data, and external threat intelligence feeds. It will employ lightweight agents deployed across the network and endpoints to collect relevant data. A high-performance stream processing engine, such as Apache Flink or Apache Kafka Streams, will ingest and process the data streams in real-time. This engine will support scalable processing of large volumes of data and enable low latency analytics.

III. ALGORITHM

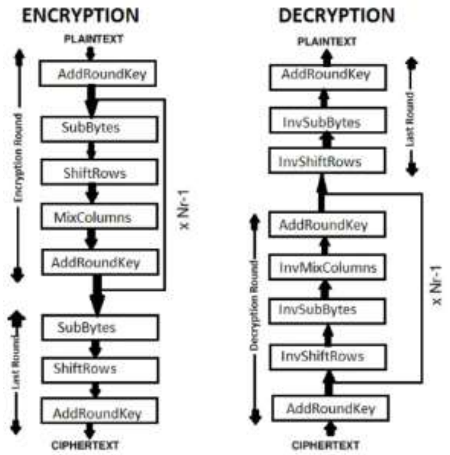
Advanced Encryption Standard (AES) is a widely adopted symmetric encryption algorithm developed to provide robust security for sensitive data. Chosen as a replacement for the aging Data Encryption Standard (DES), AES was selected by the National Institute of Standards and Technology (NIST) in 2001 after a rigorous public competition. AES operates as a symmetric cipher, meaning it employs the same key for both encryption and decryption processes, with key sizes available in 128, 192, and 256 bits. The AES algorithm supports three key sizes: 128, 192, and 256 bits. The key expansion process generates a key schedule from the original key. This schedule is used in the encryption and decryption processes. Initial Round Key Addition: The input data block is divided into a 4x4 matrix known as the state. Each byte of the state is XORed with the corresponding byte of the first round key. Rounds (for 128-bit keys, there are 10 rounds; for 192-bit keys, there are 12 rounds; for 256-bit keys, there are 14 rounds): Each round consists of four transformation stages: Sub Bytes: Each byte of the state is substituted with a corresponding value from an S-box. Shift Rows: The bytes in each row of the state are shifted cyclically. MixColumns: Each column of the state is transformed using a matrix multiplication operation. Add Round Key: The state is XORed with the round key derived from the key schedule. Final Round: The final round omits the Mix Columns stage. Output: After the final round, the resulting state is the ciphertext. Decryption: The decryption process is similar to encryption but in reverse. Instead of the encryption key schedule, the decryption key schedule is used. Each stage in the decryption process is the inverse of the corresponding stage in Encryption: Inverse SubBytes InverseShiftRows Inverse Mi x Columns AddRoundKey.

```
function AESencrypt(plaintext, key)
blocks := divideIntoBlocks(plaintext);
roundKeys = getRoundKeys(key)
for (block in blocks)
//first round
addRoundKey(roundKeys[0], block);
//intermediate rounds
for (8, 10 or 12 rounds)
subBytes(block);
shiftRows(block);
mixColumns(block);
addRoundKey(roundKeys[..], block);
//last round subBytes(block); shiftRows(block);
addRoundKey(roundKeys[numRounds - 1], block);
ciphertext := reassemble(blocks);
return ciphertext;
```

Final Output: After the final decryption round, the original plaintext is obtained. Key Schedule: The key schedule generates round keys from the original encryption key. Each round key is derived from the previous round key through a series of transformations. S-Box: The S-box is a substitution table used in the SubBytes stage. It provides a non-linear substitution of each byte of the state. Rcon: Rcon is a round constant used in the key expansion process. It provides additional entropy to each round key. This is a high-level overview of the AES algorithm. Each step involves specific mathematical operations that ensure the security and efficiency of the encryption and decryption processes.

IV. EXISTING FRAMEWORK

Detecting and notifying users of suspicious activities in real-time often involves a combination of technologies and methodologies. Here’s an outline of an existing system that could accomplish this: The system starts by collecting data from various sources. This could include network traffic logs, system logs, application logs, user activity logs, sensor data (in the case of IoT devices), and more. The data collection process may involve agents installed on endpoints, network monitoring tools, log aggregators, etc.

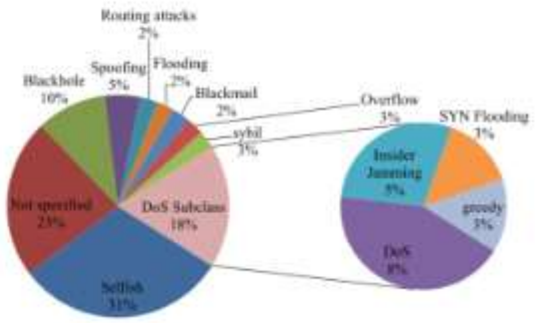


Once collected, the data is aggregated and processed in realtime. This could involve parsing log files, extracting relevant information, and structuring it in a format suitable for analysis. Tools like Elasticsearch, Logstash, and Kafka are commonly used for these purposes. Anomaly detection algorithms analyze the processed data to identify patterns that deviate from normal behavior. These algorithms could be based on statistical models, machine learning techniques (such as clustering, classification, or neural networks), or a combination of both. They continuously learn and adapt to new threats. In some cases, the system may be configured to trigger automated responses to mitigate or contain the detected threats. This could include actions like blocking malicious IP addresses, isolating compromised endpoints, or disabling user accounts. The system operates in a continuous monitoring mode, constantly analyzing incoming data streams for new threats. It also incorporates a feedback loop mechanism where the effectiveness of detection mechanisms is continuously evaluated, and the system adapts and improves over time.

In addition to anomaly detection, the system can employ rule-based detection mechanisms to flag known suspicious

activities based on predefined rules. These rules could be signatures of known threats or compliance requirements. The system correlates data from different sources to provide context to detected anomalies or suspicious activities. For example, correlating failed login attempts with unusual outbound network traffic could indicate a potential data ex-filtration attempt.

When suspicious activities are detected, the system generates alerts and notifies relevant stakeholders. Alerts can be delivered through various channels such as email, SMS, push notifications, or integration with collaboration tools like Slack or Microsoft Teams. The severity of alerts can be prioritized based on the perceived risk level. The system provides tools and interfaces for security analysts to investigate alerts further. This may include providing detailed logs, network packet captures, historical data, and contextual information to aid in the investigation process.



V. MODULE

A. Socket.IO: Utilizes Socket.IO library for enabling bidirectional, event based communication between the server and clients. Supports real-time messaging, broadcasting, and roombased communication.

B. Express.js: Leverages Express.js, a minimal and flexible Node.js web application framework, for handling HTTP requests and routing. Facilitates the creation of RESTful APIs for user authentication, message handling, and other server-side functionalities.

C. HTML, CSS, and JavaScript: Utilizes HTML for creating the structure and content of web pages, including the chat interface and user interface elements.

Employs CSS for styling and layout customization, enhancing the visual appeal and usability of the application. Incorporates JavaScript for implementing dynamic client side functionalities, such as handling user inputs, updating the chat interface in real-time, and interacting with the server via AJAX requests.

D. Real-time Messaging: Enables users to exchange text messages in real-time using Web Socket-based communication provided by Socket.IO. Supports features such as sending and receiving messages, displaying message history, and notifying users of new messages.

E. Room-based Chatting: Organizes users into chat rooms or channels based on shared interests, topics, or groups. Allows users to join, create, or leave chat rooms dynamically, facilitating group conversations and private messaging.

F. Message Encryption and Security: Implements encryption techniques to secure message transmission and protect user privacy. Utilizes encryption algorithms such as AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman) for end-to-end encryption (E2EE) of messages.

G. Responsive Design: Adopts responsive design principles to ensure compatibility and optimal user experience across various devices and screen sizes. Utilizes CSS frameworks like Bootstrap or Tailwind CSS for building responsive and mobile-friendly user interfaces.

H. Data Persistence: Integrates with databases (e.g., MongoDB, MySQL) to store user data, chat logs, and other application-related information.

VI. METHODOLOGY

Identify and collect relevant data sources, including network traffic logs, system logs, application logs, user activity logs, and external threat intelligence feeds. Implement mechanisms for real-time data collection and aggregation from these sources. Ensure compliance with data privacy regulations and organizational policies during data collection. Implement anomaly detection algorithms to analyze the collected data streams and identify deviations from normal behavior. Choose appropriate techniques such as statistical analysis, machine learning models, or rule-based approaches based on the nature of the data and the types of suspicious activities being monitored. Continuously refine and update the anomaly detection models based on feedback and new threat intelligence. Develop and implement predefined rules or signatures to detect known suspicious activities or indicators of compromise (IOCs). These rules can be based on patterns commonly associated with malicious behavior, such as specific network traffic patterns, file access patterns, or user authentication events. Integrate data correlation mechanisms to provide context to detected anomalies and alerts. Correlate data from multiple sources to identify potential cause-and-effect relationships and prioritize alerts based on their significance. Incorporate contextual information such as user roles, asset sensitivity, and historical behavior patterns into the analysis process. Design an alerting mechanism to notify users and stakeholders in real-time when suspicious activities are detected. Define escalation procedures for escalating alerts based on severity levels or specific criteria. Implement various notification channels such as email, SMS, push notifications, or integration with collaboration tools. Developing a methodology for detecting and notifying users of suspicious activities in real-time involves a systematic approach to ensure robust threat detection and response capabilities. Initially, the scope and objectives of the detection system are defined, alongside identifying the specific types of suspicious activities to monitor and the relevant stakeholders to notify. Data collection mechanisms are then established, encompassing diverse sources such as network traffic logs, system logs, and user activity logs, ensuring real-time aggregation and adherence to data privacy regulations. Anomaly detection algorithms, incorporating statistical analysis, machine learning models, or rule-based approaches, are implemented to analyze data streams and detect deviations from normal behavior. Additionally, predefined rules or signatures are employed to detect known suspicious activities, augmented by contextual analysis to prioritize alerts effectively. When suspicious activities are detected, alerts are triggered in real-time, leveraging various notification channels such as email, SMS, or push notifications. Response plans are formulated to mitigate threats, including automated actions like blocking malicious IP addresses or isolating compromised endpoints. Continuous monitoring, evaluation, and refinement of the system ensure ongoing effectiveness, complemented by training programs to promote security awareness within the organization. Documentation and reporting mechanisms capture system configurations, detected threats, and response actions taken, facilitating accountability and informed decision making. Through this comprehensive methodology, organizations can establish a proactive defense against cyber threats, fostering a secure environment for their operations and stake holders.

VII. CONCLUSION

In conclusion, the implementation of real-time detection and notification systems for suspicious activities represents a pivotal advancement in ensuring the security and integrity of digital platforms. By leveraging cutting-edge technologies and proactive monitoring strategies, such systems empower organizations to swiftly identify and respond to potential threats, mitigating risks before they escalate. Through continuous refinement and adaptation, these solutions offer a robust defense mechanism against evolving cyber threats, safeguarding user data, privacy, and overall trust in the digital ecosystem. As we continue to prioritize security in an increasingly interconnected world, the development and deployment of such systems are paramount in fortifying our collective resilience against malicious actors. The implementation of real-time detection and notification systems for suspicious activities is not just a technological advancement but a fundamental necessity in today's digital landscape. With cyber threats evolving at an unprecedented pace, organizations face constant pressure to stay ahead of malicious actors who seek to exploit vulnerabilities for their gain. By harnessing the power of advanced algorithms, machine learning, and artificial intelligence, these systems enable organizations to detect anomalies and unauthorized activities in real-time, providing invaluable insights into potential security breaches. The significance of real-time notification cannot be over stated. Timely alerts allow security teams to take swift action, mitigating risks and minimizing the potential impact on users and critical infrastructure. Whether it's detecting unusual login attempts, suspicious network traffic, or unauthorized access to sensitive data, real-time notification systems serve as the frontline defense, helping organizations stay one step ahead of cyber threats. The proactive nature of these systems not only enhances security but also instills confidence among users. By demonstrating a commitment to protecting user privacy and data integrity, organizations foster trust and credibility, essential components of any successful digital platform or service. In an age where data breaches and cyber-attacks dominate headlines,

investing in robust detection and notification mechanisms is not just a best practice but a strategic imperative. As we look to the future, the importance of real-time detection and notification systems will only continue to grow. With the proliferation of IoT devices, cloud-based services, and interconnected networks, the attack surface expands, presenting new challenges for security professionals. However, by embracing innovative technologies and adopting a proactive mindset, organizations can strengthen their security posture and effectively combat emerging threats.

REFERENCE

R. R. Chandan and P. K. Mishra, "A review of security challenges in adhoc network," Int. J. Appl. Eng. Res., vol. 13, no. 22, pp. 16117–16126, 2018.

O. H. Younis, S. E. Essa, and E. S. Ayman, "A survey on security attacks/defenses in mobile ad-hoc networks," Commun.

Appl. Electron., vol. 6, no. 10, pp. 1–9, Apr. 2017.

G. Keerthana and P. Anandan, "A survey on security issues and challenges in mobile ad-hoc network," EAI Endorsed Trans. Energy Web, vol. 5, no. 20, Sep. 2018, Art. no. 155743 W. Bouassaba, A. Nabou, and M. Ouzzif, "Review on machine learning based intrusion detection for MANET security," in Proc. 9th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM), Oct. 2022, pp. 1–6.

D. G. Kampitaki and A. A. Economides, "Selfishness in mobile adhoc networks: A literature review on detection techniques and prevention mechanisms," IEEE Access, vol. 11, pp. 86895–86909, 2023.

H. Shah, V. Kakkad, R. Patel, and N. Doshi, "A survey on game theoretic approaches for privacy preservation in data

mining and network security," Proc. Comput. Sci., vol. 155, pp. 686–691, Jan. 2019.

P. Dasgupta and J. B. Collins, "A survey of game theoretic approaches for adversarial machine learning in cybersecurity

tasks," AI Mag., vol. 40, no. 2, pp. 31–43, Jun. 2019.

K. Merrick, M. Hardhienata, K. Shafi, and J. Hu, "A survey of game theoretic approaches to modelling decision-making in information warfare scenarios," Future Internet, vol. 8, no. 3, p. 34, Jul. 2016.

X. Wang, Y. Wang, J. Liu, Z. Huang, and P. Xie, "A survey of game theoretic methods for cyber security," in Proc. IEEE 1st Int. Conf. Data Sci. Cyberspace (DSC), Jun. 2016, pp. 631–636.

C. Kiennert, Z. Ismail, H. Debar, and J. Leneutre, "A survey on game theoretic approaches for intrusion detection and response optimization," ACM Computer. Survey., vol. 51, no. 5, pp. 1–31, Sep. 2019.

A. Sharah, M. Alhaj, and M. Hassan, "Selfish dynamic punishment scheme: Misbehavior detection in MANETs using cooperative repeated game," Int. J. Comput. Sci. Netw. Security vol. 20, no. 3, pp. 168–173, 2020.

R. F. Olanrewaju, F. Anwar, R. N. Mir, M. Yaacob, and T. Mehraj, "Bayesian signaling game based efficient security model for MANETs," in Proc. Future Inf. Commun. Conf vol. 2. Cham, Switzerland: Springer, 2020, pp. 1106–1122