Internal Intrusion Detection and Protection by Self-Monitoring via Forensic Techniques and help of Data Mining

[1st] Prof Shah S.N, [2nd]Dr. Taware G. G, [3rd]Shreyash Bhandwalkar, [4th]Vaishnavi Ghadge, [5th]Komal Papal
[1]*HOD, Department of Computer Engineering, Sharadchandra Pawar college of Engineering and Technology Someshwarnagar Pune, India*
[2]*Assistant Professor, Department of Computer Engineering, Sharadchandra Pawar college of Engineering and Technology Someshwarnagar Pune, India*
[3,4,5] *Student, Department of Computer Engineering, Sharadchandra Pawar college of Engineering and Technology Someshwarnagar Pune, India*

***Abstract*— Nowadays, billions of individuals around the world rely on the internet for daily activities. With this increasing reliance comes a growing need for advanced cybersecurity solutions. One such emerging technology is intrusion detection, which plays a vital role in identifying and preventing malicious actions within a system. This project introduces a new generation of security technology known as the Intrusion Detection and Protection System (IDPS), which continuously monitors user behavior across a network using a localized grid-based process. The system is designed to detect suspicious activities and respond effectively by analyzing behavioral patterns and building user profiles for real-time monitoring. To validate the effectiveness of the proposed system, it is assessed using both traditional intrusion detection systems and modern forensic analysis methods. The foundational study also includes a comprehensive literature review of various Intrusion Detection Systems (IDS) and Internal Intrusion Detection Systems (IIDS), each leveraging distinct algorithms and data processing techniques to detect intrusions in real time. The Internal Intrusion Detection System (IIDS), specifically developed during this research, utilizes pre-established algorithms to identify and differentiate between legitimate and unauthorized user activities within a network environment. This approach aims to enhance cyber analytics by offering more precise and timely threat detection.**

I. INTRODUCTION

In recent decades, portable computing systems have become increasingly popular, offering users more accessible and convenient ways to manage their daily activities. However, as the functionality and capabilities of these systems continue to evolve, security has emerged as a major concern. Cyber attackers often target these devices to carry out malicious actions such as corporate espionage, disabling essential processes, or even completely destroying critical operations.

Among the many well-known cyber threats—such as spear-phishing, eavesdropping, DDoS (Distributed Denial of Service) attacks, and pharming—executive-level attacks remain particularly difficult to detect due to the limitations of conventional security tools like firewalls.

Traditionally, intrusion detection systems (IDS) are designed to guard against external threats. Most current systems rely heavily on basic authentication methods, such as verifying a user ID and password at login. However, attackers can deploy Trojan horses to capture login details or use brute-force methods like dictionary attacks to uncover passwords. Once successful, they can gain unauthorized access to sensitive files, alter system settings, or compromise user accounts entirely.

Host-based security solutions typically combine network-level intrusion detection with standard monitoring practices, but distinguishing legitimate users from attackers remains a challenge—especially when threats come from insiders or when attackers use valid login credentials. Attack packets often carry spoofed IP addresses, making the origin difficult to trace. This allows an attacker with a seemingly valid login to bypass basic security filters. Computer forensics plays a crucial role in identifying and analyzing digital evidence during security incidents. It treats computer systems as digital crime scenes, investigating malicious activities such as malware deployment, virus

propagation, and DDoS attacks. Many existing intrusion detection techniques focus on identifying unusual network behaviors or spotting known attack patterns by analyzing log files. While these tools enhance overall cyber security, they often fall short when it comes to detecting sophisticated intrusions—especially when attackers use valid credentials or operate remotely.

In earlier research, we introduced a security system that applies data mining and digital forensic techniques at the command level (rather than the system call level) to capture user behavior and identify potential threats. This approach is particularly useful when attackers launch attacks over multiple sessions, making detection more complex.

Current host-based security architectures and signature-based intrusion detection systems can monitor and flag known threats in real-time. However, identifying the actual perpetrator remains challenging, especially when attack kits are bundled with forged digital identities or when intrusions occur using legitimate access patterns. Monitoring large volumes of system calls (SCs) and applying behavioral mining techniques can provide deeper insight into both malicious users and legitimate client behavior—especially when system-level data is leveraged effectively for threat detection.

II. RELATED WORK

Computer forensics is a specialized field focused on identifying, preserving, recovering, analyzing, and presenting digital evidence related to security incidents. It treats digital systems as potential crime scenes, investigating malicious activities such as Distributed Denial of Service (DDoS) attacks, the spread of malware, viruses, and other harmful code.

Most intrusion detection techniques aim to pinpoint abnormal network behavior and extract distinct traits from malicious data packets. These logs and files often contain traces of past cyber attacks, making it possible to simulate and study attack patterns using synthetic log files. Through comprehensive reviews and comparisons of various intrusion detection methods, researchers have been able to highlight current challenges in the field, offering valuable insights for future development.

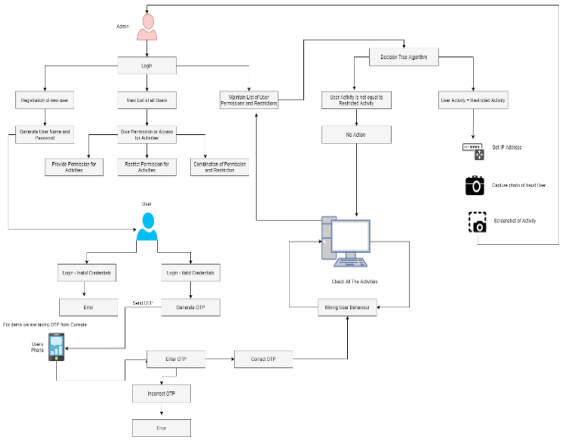
While these detection methods significantly improve cyber security, they face challenges when dealing with complex attack strategies—particularly multi-session or multistage attacks. In such cases, attackers launch coordinated attacks over time, which are harder to trace. To address this, some systems utilize an Intelligent Intrusion Detection and Prevention System (IIDPS) that leverages forensic profiling and data mining to track user behavior and identify coordinated threats in real-time.

Although these approaches show promise, earlier work did not address system call (SC) filtering—an important layer of security. Another innovative solution integrates computer forensics with a knowledge-based system. This system operates on a predefined model that allows specific SC sequences, using these patterns to detect and restrict unauthorized program activity. It can quickly identify suspicious SC chains that match known attack sequences stored in its knowledge base, often recognizing threats within just two seconds with minimal computational load.

Additionally, a dynamic, grid-based intrusion detection platform has been proposed. This system taps into the extensive computing capabilities of a grid infrastructure to adaptively monitor and analyze large volumes of attack packets in real-time. One significant application of such security solutions is in smart card technology, particularly within advanced metering infrastructure (AMI) systems. These smart meters enable two-way communication between users and utilities, necessitating strong protection mechanisms.

The authors highlight that encryption, authentication, and authorization form the first layer of security in AMI systems. Intrusion Detection Systems (IDS) are considered the second line of defense, providing critical support in identifying threats that bypass initial protective measures.

III. PROPOSED SYSTEM



The Internal Intrusion Detection and Protection System (IIDPS) is a security solution designed to identify and counteract malicious actions targeting a system at the Supervisor Call (SC) level. The system utilizes advanced rhetorical identification methods and data analysis to study Supervisor Call instruction patterns (SC patterns), as described below. These SC sequences, often the longest and most frequently appearing in a user's log file, provide key insights into user behavior.

These SC sequences are uniquely tied to individual users, with specific patterns that are rarely found in the actions of others. By analyzing these sequences, the IIDPS can recognize suspicious behavior that may signal an attempt to compromise the system. The system focuses on identifying and mitigating harmful actions by analyzing the SCs and their associated patterns.

Key features of the proposed system include:

- Reliable Intrusion Detection: The primary goal is to deliver a trustworthy and effective intrusion detection solution, ensuring minimal false alarms while identifying real threats.
- Continuous Monitoring and Self-Analysis: The system employs a self-analysis approach, constantly tracking user activities to maintain a detailed log of their typical actions and behaviors.
- Utilizing Data Mining and Forensic Techniques: Internal system calls (SCs) are analyzed using data mining methods to uncover potential intrusion attempts. Forensic techniques help track any suspicious activity, providing deeper insights into potential threats.
- SC Pattern Recognition for User Profiling: The ability to recognize and document specific SC patterns associated with each user allows the system to differentiate between routine and abnormal actions.
- Excluding Normal Activity: Routine and everyday actions are disregarded in favor of detecting less frequent, potentially suspicious activity that may indicate an intruder.
- Reporting Suspicious Activity: If the system identifies limited or unusual activity, it promptly alerts the appropriate authorities, triggering an investigation to determine if an attack is underway.

The IIDPS is designed to offer proactive protection by continuously analyzing user SC patterns and detecting anomalies that may indicate an intrusion, ensuring the system remains secure.

IV. METHODOLOGY

This section introduces the architecture of the Intelligent Intrusion Detection and Prevention System (IIDPS) and provides a detailed overview of its core components. The IIDPS enhances the accuracy of intrusion detection by analyzing system calls (SCs) to uncover the forensic traits of individual users. When deployed in a parallel computing environment, the system can significantly reduce detection response time, offering faster and more effective protection against insider threats.

By identifying malicious commands issued by a user, the IIDPS can actively prevent intrusions and protect critical systems. The system is composed of several key elements: a local computing grid, a mining server, a detection server, an SC monitor and filter, and three dedicated repositories for storing user log files, user behavior profiles, and attacker profiles.

The SC monitor and filter is implemented as a kernel module within the operating system. It captures SCs submitted to the kernel, recording them in a structured format—(uid, pid, SC)—where uid represents the user ID, pid is the process ID, and SC is the system call invoked. These sequences, along with user input, are stored in individual user log files to track the order and pattern of SCs issued.

The mining server utilizes data mining algorithms to analyze these logs and uncover patterns in user behavior. These behavioral signatures are then compiled into detailed user profiles. Simultaneously, the detection server evaluates real-time user activity

against both known attacker patterns and the behavioral profiles of legitimate users to detect anomalies or unauthorized behavior.

If a potential threat is identified, the detection server alerts the SC monitor and filter, which immediately isolates the user from the system to prevent further damage. Both the detection and mining servers run on a local grid to support the system's ability to perform real-time analysis and pattern recognition at scale.

An added layer of user verification is incorporated when a login attempt uses credentials that do not align with typical behavior. By comparing current SC sequences to those stored in various user profiles, the IIDPS can accurately infer the true identity of the active user, even if valid credentials were used by an impostor.

The SC monitor and filter also enforces role-based restrictions through a "class-limited-SC list," which defines system calls that are not permitted for specific user roles. For instance, a secretary would not have access to privileged SCs reserved for administrators. These restrictions ensure that users operate strictly within their designated access levels, enhancing the system's ability to prevent privilege escalation and unauthorized command execution.

A decision tree addresses classification problems by utilizing a tree-based structure. In this model, each internal node denotes a specific attribute or feature, while each leaf node corresponds to a class label. Decision trees are capable of representing any Boolean function involving discrete input attributes.

The pseudocode below outlines the decision tree algorithm implemented in our approach.
Algorithm: Decision Tree (User Activity, target Attribute, feature List)
Input:

- *User Activity* – A dataset consisting of user behavior or activity records
- *targetAttribute* – The label or category we aim to predict
- *featureList* – A collection of features used for making decisions

Output:

- A constructed decision tree

Steps:

1. If all instances in *User Activity* belong to a single class:
   → Return a leaf node labeled with that class.
2. If *feature List* is empty:→ Return a leaf node labeled with the most frequent class in *User Activity*.
3. Choose the most informative feature *A* from *feature List* (e.g., based on Information Gain or another impurity metric).
4. Create a new internal node that splits on feature *A*.
5. For every possible value *v* that *A* can take:a. Partition *User Activity* into a subset *S* where feature $A = v$
   b. If *S* is empty:
   → Attach a leaf node labeled with the majority class in the original dataset.
   c. Otherwise:
   → Recursively apply the DecisionTree algorithm on subset *S*, the same target attribute, and the remaining features (*featureList* – *A*).
   → Attach the resulting subtree to the branch where $A = v$.
6. Return the completed decision tree.

V. RESULT AND DISCUSSION

To determine the accuracy of the Intrusion Detection System (IDS), an evaluation was carried out involving four different user profiles. Four separate users were configured with access limitations on specific restricted actions. Each of them logged in and performed a variety of actions, including both allowed and restricted ones, to assess how effectively the system detects prohibited activities.

Logged in Users	Performed All Activities	Performed Normal Activities	Expected Ignored Activities	Actual Ignored Activities	Performed Restricted Activities	Expected Capture Activities	Actual Captured Activities
User 1	150	148	148	148	2	2	2
User 2	100	99	99	99	1	1	1
User 3	70	70	70	70	0	0	0
User 4	30	27	27	27	3	3	3
Total	350	344	344	344	6	6	6

- **Performed All Activities** → Total activities done by the user
- **Performed Normal Activities** → Standard activities executed
- **Ignored Activities** → Expected and actual ignored tasks
- **Performed Restricted Activities** → Activities that were limited
- **Captured Activities** → Activities that required recording

In total, the users performed 350 actions, summarized in the following breakdown:

- User 1 executed 150 actions, of which 148 were permitted and went undetected by the

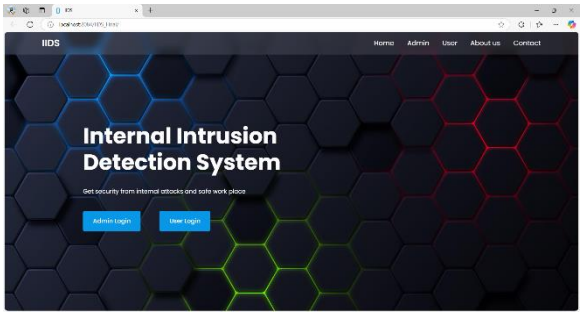
system, while 2 were restricted and successfully flagged by the system.

- User 2 completed 70 actions—all of which were non-restricted—so the system had nothing to capture.
- User 3 carried out 100 activities, 99 of which were allowed. Only one was restricted and was properly identified.
- User 4 performed 30 actions, with 27 being normal and 3 being restricted, all of which were detected.

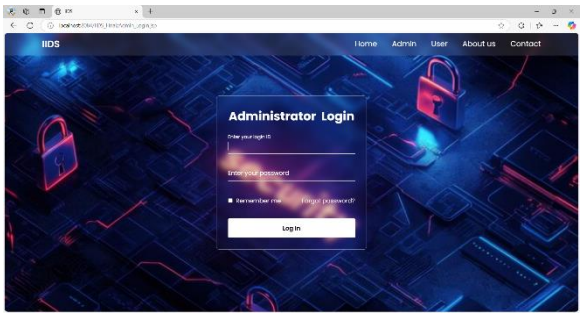
From this testing, the system's accuracy was calculated:

- Accuracy for restricted activity detection = (Number of correctly captured restricted actions / Total restricted actions expected) × 100
  → 6 / 6 × 100 = 100%
- Accuracy for ignoring normal activity = (Correctly ignored normal actions / Total normal actions) × 100
  → 344 / 344 × 100 = 100%

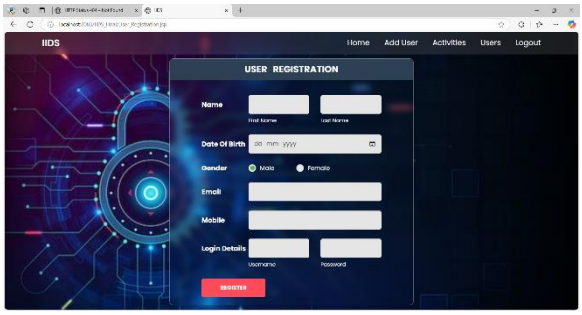
II. RESULT



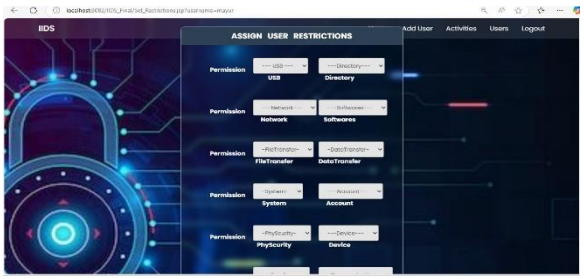
This is the IIDS web application's home page. Our application page will offer admin and system user login choices as soon as it launches.



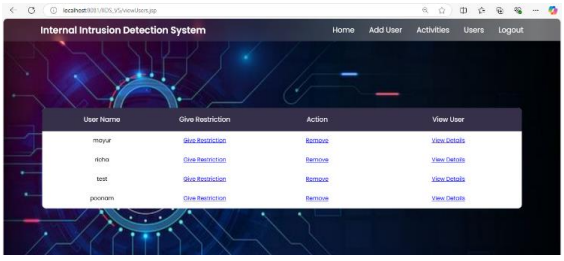
Here Admin User is logging to perform further activities. Once the admin has logged in, they will register new users



Here, we have a form designed to collect the new user's information. The administrator creates the login credentials, which the additional user can use to access the organization's work after being added.



After a new user has registered, the administrator can see a list of all users and choose which one to use to limit specific behaviour. We have put limitations for the USB connection and confidential folder for the demo. Each user just needs to set this once (restriction/non-restriction).



In addition to restrictions, the administrator has the ability to view a specific user's details, remove a user's entry, and update user details.

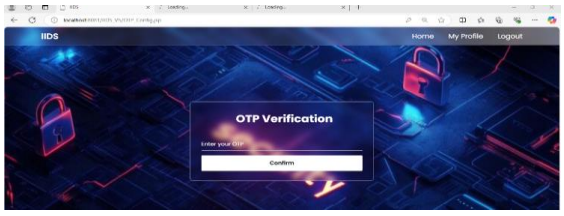
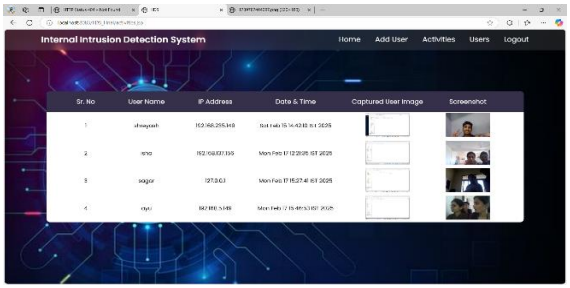


Fig 8.6: OTP Generation

Moving on to user login, we have created a user login screen and added an additional module for increased security, generates a one-time password that the user will receive when entering their valid username and password and after entering the right OTP.



Following successful login, the user can engage in typical activities, such as those that are not prohibited by the administrator or those that are. When a user engages in restricted activity, our system will automatically detect the attached and begin recording the information. If the user engages in non-restricted activity, our self-monitoring system will not intervene and will keep tracking the activities. To prove that the user has engaged in restricted activity, our system will first launch the system camera, take a picture of them, and then take a screenshot of the current screen. Following the capture of a screenshot and snapshot, the system will obtain the connected IP address, the date and time of the attack, and forward all of this data to the administrator. The administrator will obtain the attacker's details, as seen in the screenshot above.

VIII. ADVANTAGES

1. Improved Security with Layered Authentication:
The system employs multiple layers of verification to strengthen user authentication, making it harder for unauthorized individuals to gain access.
2. Immediate Detection of Unusual Behavior:
It monitors user activity in real-time, enabling the system to instantly flag and respond to suspicious or abnormal actions.
3. Fast and Efficient Incident Handling:
The architecture allows for rapid identification and containment of security breaches, minimizing potential damage.
4. Minimized Risk from Insider Attacks:
By analysing user behaviour and restricting access based on roles, the system effectively reduces the chances of internal misuse.
5. Detailed Forensic Logging for Investigation:
All actions are logged systematically, allowing for thorough evidence gathering and post-incident analysis to support accountability and future prevention.

IX. CONCLUSION

This study emphasizes the critical need to address internal security threats, which are often overlooked compared to external attacks. Traditional security solutions primarily focus on preventing breaches from outside, leaving systems vulnerable to malicious actions from authorized users. The Internal Intrusion Detection and Protection System (IIDPS) aims to bridge this security gap by integrating forensic techniques and behavioural analysis to monitor user activities in real time. By identifying abnormal behaviour patterns, the IIDPS can swiftly detect and block insider threats before they cause damage. Its ability to provide rapid responses enhances overall system security. Moving forward, further advancements could improve its detection accuracy and adaptability across various computing environments.

REFERENCES

[1] A Self-Attention-Based Deep Convolutional Neural Networks for IIoT Networks Intrusion Detection.
[2] DTITD: An Intelligent Insider Threat Detection Framework Based on Digital Twin and Self-Attention Based Deep Learning Models.
[3] Hunt for Unseen Intrusion: Multi-Head Self-Attention Neural Detector.
[4] Network Intrusion Detection Method Based on CNN-BiLSTM-Attention Model.
[5] S. Li, G. Chai, Y. Wang, G. Zhou, Z. Li, D. Yu, and R. Gao, “CRSF: An intrusion detection framework for industrial Internet of Things based on pretrained CNN2D-RNN and SVM,” IEEE Access, vol. 11, pp. 92041–92054.
[6] Y. Zhang, C. Yang, K. Huang, and Y. Li, “Intrusion detection of industrial Internet-of-Things based on reconstructed graph neural networks,” IEEE Trans. Netw. Sci. Eng., vol. 10, no. 5, pp. 2894–2905.
[7] M.Mohy-eddine, A.Guezzaz, S. Benkirane, and M. Azrour, “An effective intrusion detection approach based on ensemble learning for IIoT edge computing,” J. Comput. Virol. Hacking Technology.
[8] M.Nuaimi,L.C.Fourati, and B.B.Hamed,“Intelligent approaches toward intrusion detection systems for industrial Internet of Things: A systematic comprehensive review,” J. Netw. Comput. Appl.

[9] M. Tanveer and S. Shabala, “Entangling the interaction between essential and nonessential nutrients: Implications for global food security,” in Plant Nutrition and Food Security in the Era of Climate Change. Amsterdam, The Netherlands: Elsevier

[10] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, “Blockchain for the IoT and industrial IoT: A review,” Internet Things.