International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT25112861

Security and Identification of Internal Intrusions by Self-Monitoring

Sharayu Kadam*, Radhika Bhoite

*Department of Information Technology, Pune, Maharashtra, India

ARTICLEINFO

Article History:

Accepted : 01 May 2025

Published: 03 May 2025

Publication Issue

Volume 11, Issue 3

May-June-2025

Page Number

08-13

ABSTRACT

The increasing reliance on the internet by billions worldwide has made cyber security a vital concern. Among emerging security solutions, intrusion detection technologies have gained importance for their ability to monitor systems and prevent unauthorized activities. These systems use localized network mechanisms to continuously observe user behaviors and identify potential threats over time. To enhance system security, this project introduces a novel framework known as the Call-Level Intrusion Detection and Protection System, which creates and updates user behavior profiles to detect anomalies effectively. The performance of the proposed model has been validated through the use of forensic analysis and established intrusion detection methodologies. Additionally, this research reviews prior studies focused on Intrusion Detection Systems (IDS) and Internal Intrusion Detection Systems (IIDS). Building upon this foundation, a new IIDS model was developed, utilizing advanced algorithms to differentiate between legitimate and malicious activities within network infrastructures.

Keywords: System calls, malicious activity, intrusion detection, and internal attacks

Introduction

Over the past few decades, portable computer systems have been widely adopted to offer consumers easier and more direct access to technology in their daily lives. However, as these devices become increasingly powerful and sophisticated, security concerns have also grown significantly. Portable computers have become prime targets for attackers aiming to breach systems, steal sensitive corporate information, disrupt critical operations, or even completely disable them. Among the many forms of cyber-attacks including spear-phishing, eavesdropping, distributed denial-of-service (DDoS) attacks, and pharming—executive-level attacks are especially difficult for conventional security tools like firewalls to detect.

Typically, Intrusion Detection Systems (IDSs) are designed to protect against external threats. Most systems today rely heavily on basic authentication

methods such as user IDs and passwords. However, attackers often bypass these measures by using Trojan horses to steal login credentials or by executing extensive dictionary attacks to crack passwords.

Many host-based security solutions currently integrate network-based intrusion detection methods that attempt to minimize disruptions while still recognizing potential intrusions. Nevertheless, identifying sophisticated threats remains a difficult task, especially when attackers utilize valid login credentials and genuine IP addresses to mask their activities. Verifying a user's identity based solely on an ID and password proves insufficient against such sophisticated breaches.

The field of computer forensics plays a crucial role in addressing these challenges by aiming to collect, analyze, preserve, and present digital evidence related to security incidents. Similar to investigating a physical crime scene, forensic analysts examine system activities, including DDoS attacks and the distribution of malicious software like viruses, malware, and worms. Most traditional intrusion detection approaches rely heavily on analyzing historical log files, aiming to detect malicious network behavior and identify common patterns in attack signatures. While these methods have strengthened overall network security, they often fall short in scenarios where attackers log in remotely with stolen but valid credentials.

In our prior work, we introduced a security model that shifts focus from the system call (SC) level to the command level, utilizing data mining and forensic methods to gather detailed user behavior information. This approach enables a more refined analysis, particularly when attacks are carried out over multiple sessions.

Modern, behavior-based intrusion detection systems and host-based security architectures are increasingly capable of recognizing known attack patterns. However, detecting intrusions remains a significant challenge, especially since attackers frequently exploit legitimate login information or deploy advanced automated attack tools. Although monitoring operating system-level system calls (SCs) provides valuable insights for detecting malicious behavior, effectively managing and analyzing the vast number of SCs generated remains a key hurdle in building highly efficient security systems.

METHODOLOGY

This section introduces an overview of the IIDPS framework, followed by a detailed explanation of each of its components. By implementing the IIDPS on a parallel computing setup, it further reduces response time for threat detection, making it highly effective against insider threats. IIDPS works by identifying malicious activities and blocking attacks before they can impact the protected environment.

The core structure of IIDPS consists of three main repositories—user log files, user profiles, and an attacker profile—as well as a mining server, a detection server, a system call monitor (SC monitor), and a local computational grid. Within the protected system, the SC monitor and filter act as a loadable kernel module that captures and records system calls submitted to the kernel. It stores the data in the format (uid, pid, SC), where uid represents the user ID, pid denotes the process ID, and SC indicates the system call made by the user. The SC monitor also saves the sequence of SCs in a user log file, preserving the order in which actions were performed.

The mining server processes these logs using data mining techniques to uncover behavioral patterns and usage habits, which are then documented within individual user profiles. Simultaneously, an attacker profile is constructed by collecting known malicious SC patterns.

The detection server continuously monitors user behavior by comparing live SC patterns against both the legitimate user profiles and the attacker profile. If suspicious behavior is detected, the detection server alerts the SC monitor and filter, which immediately isolates the offending user from the protected system, preventing further damage. Both the mining server

and the detection server operate within the local computational grid to improve real-time detection and analysis capabilities.

When a user logs in, IIDPS compares the incoming SC patterns against stored profiles to verify user identity. If a user's current system calls do not align with their recorded behavior patterns, the system can recognize unauthorized access, even when valid credentials are used. Furthermore, the SC monitor maintains a class-restricted SC list, which specifies commands that certain user groups are prohibited from executing. For instance, secretarial staff may be restricted from submitting privileged system calls. This mechanism adds an extra layer of security by ensuring that users cannot perform actions beyond their designated permissions.

EXISTING SYSTEM

Currently, various information security strategies are employed to safeguard systems from threats such as viral attacks, unauthorized access, data tampering, and destruction. One of the primary defenses is the firewall, designed to block unwanted external access and protect internal networks from internet-based threats. However, a significant limitation of firewalls is that they primarily monitor external traffic, making it difficult to detect threats originating from within the network. Firewalls restrict cross-network access to prevent breaches but can struggle to recognize internal malicious activities.

CCTV Monitoring:

While CCTV cameras are useful for observing physical activities and identifying individuals in a location, they are ineffective in monitoring internal system operations and user behavior on digital platforms.

Limitations of the Existing System:

- **Lower detection accuracy:** Current solutions often fail to precisely identify security breaches.
- **Difficulty in detecting dangerous user behavior:** Harmful actions by legitimate users may go unnoticed.
- **Inefficient tools for identifying malicious users:** Existing detection mechanisms are not optimized for internal threat recognition.

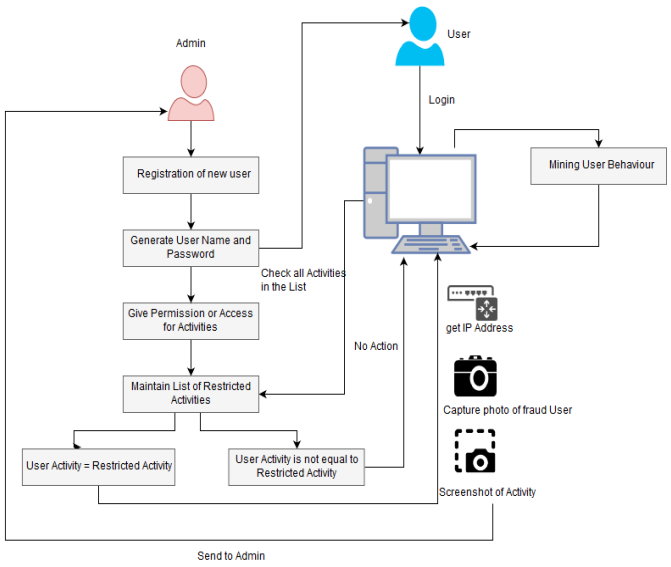
PROPOSED SYSTEM

The proposed security solution, the **Internal Intrusion Detection and Protection System (IIDPS)**, focuses on detecting malicious behaviour at the **system call (SC) level**. IIDPS employs data mining and linguistic identification algorithms to analyse patterns within supervisor call instructions (SC patterns). Given that certain call instruction sequences frequently appear in a user's system log files, IIDPS characterizes user-specific SC patterns. These patterns, which occasionally emerge across multiple user activities but are rarely found among others, help in distinguishing legitimate behaviour from malicious actions.

To effectively detect and prevent attacks, the system examines both the SCs and the SC-patterns associated with executed commands, identifying any harmful activities that could compromise the protected system.

Features of the System:

- **Self-analysis mechanism:** Continuously monitors user activities to identify unauthorized actions and potential threats.
- **Application of data mining and forensic techniques:** Enhances intrusion detection by analysing system calls internally.
- **User behaviour profiling:** Tracks SC tendencies to accurately recognize individual users and distinguish between normal and malicious activities.
- **Focused monitoring:** Regular user activities are disregarded, ensuring that only suspicious behaviour triggers alerts.
- **Timely notification:** In cases where even minimal suspicious activity is detected, the system promptly alerts the concerned authorities for further investigation.



Process Overview:

Step 1: Assume the user logging into the system is **U**. There are multiple users, namely **U1, U2, Un**, who are comparable or related to user U in their access rights or roles.

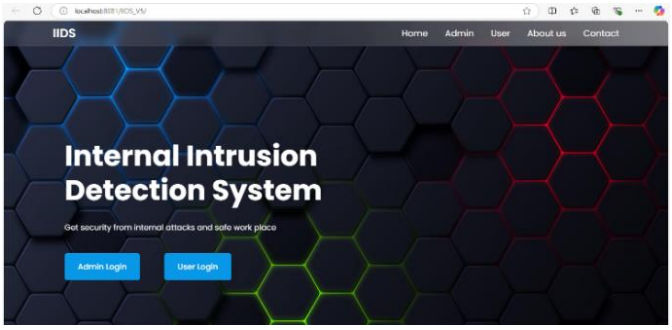
Step 2: The system, referred to as **S**, is responsible for generating and providing a **One-Time Password (OTP)** to authenticate the user's login session.

Step 3: Once authenticated, the user may attempt various actions such as installing unauthorized software, transferring files from secured locations to less secure areas, connecting external devices (e.g., USB drives), or performing other operations that violate administrator-defined policies. To detect such behaviours, the system continuously monitors the log files generated during user activities.

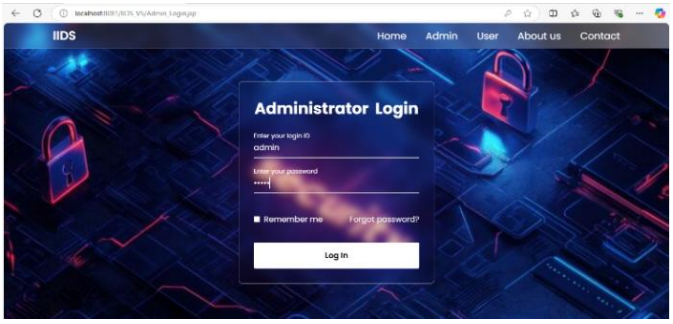
Step 4: System **S** reads the user's log files and correlates infrequent or suspicious activities with a predefined **attack list (A)**, utilizing a **detection module (D)** to identify any unauthorized access or prohibited actions.

Step 5: Upon detecting malicious behaviour, System **S** will immediately trigger alerts by capturing a **screenshot** of the activity, taking a **photo of the user** (if a camera is available), and recording the **IP address** of the machine involved, ensuring complete forensic documentation of the incident.

RESULTS UI SCREENS



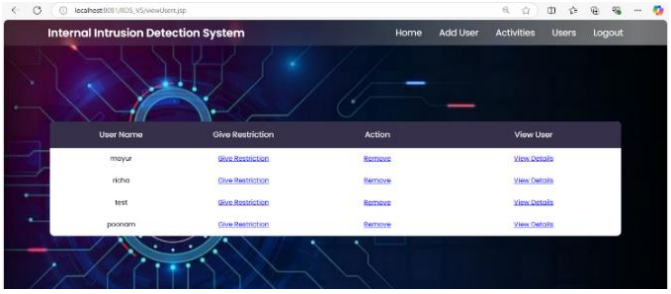
The home page of the IIDS web application serves as the initial interface for users. Upon opening the application, it provides two distinct login options one designated for administrators and the other for system users.



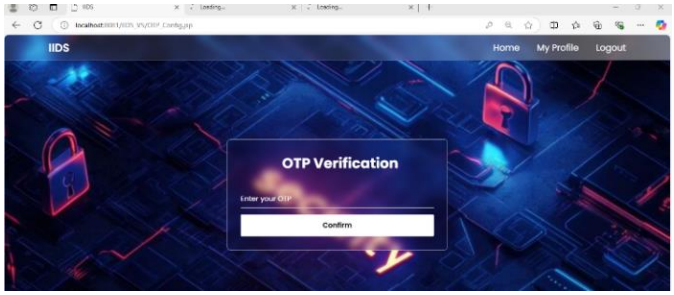
At this stage, the administrator logs into the system to carry out further operations. After successful authentication, the admin proceeds to register new users within the application.

A registration form is provided to gather the necessary details of the new user. The administrator is responsible for creating the login credentials, which the newly added user can later use to access the organization's resources.

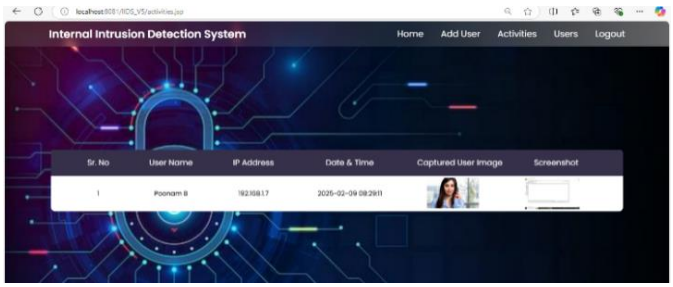
Once a new user has been registered, the administrator can view a complete list of all users and select specific users to apply behaviour restrictions. For demonstration purposes, we have implemented restrictions on USB device connections and access to confidential folders. These settings—whether to restrict or allow—only need to be configured once for each user.



Along with setting restrictions, the administrator can also view detailed information for individual users, update their records, or remove a user's entry from the system.



Proceeding to the user login process, a dedicated login screen has been developed. To enhance security, an additional module for multi-factor authentication has been integrated. Once users enter their valid username and password, a one-time password (OTP) is generated and sent to them. Access is granted only after the correct OTP is entered.



Once logged in, the user can perform various activities, including both permitted and restricted actions as defined by the administrator. When the system detects a restricted activity, it automatically initiates a recording process. If the activity is permitted, the self-monitoring system remains passive while continuing to track user actions.

To confirm that the user has engaged in restricted behaviour, the system will activate the camera to capture an image of the user and take a screenshot of the current screen. After capturing these details, the system will also record the connected IP address, along with the date and time of the incident, and send this information to the administrator. The administrator will then have access to the attacker's details as shown in the captured data.

RESULTS AND DISCUSSION

To evaluate the accuracy of our Intrusion Detection System (IDS), we conducted tests using four different users. For each user, we configured restrictions on specific activities. After setting up the restriction rules, the four users logged in and performed a series of activities, which included both normal and restricted actions. We monitored the system's ability to detect restricted activities during these sessions. In total, we carried out 350 activities across all four users.

"The following describes the activities performed by each user during the test:

- **User 1**: This user completed a total of 150 activities. Of these, 148 were non-restricted activities, which were ignored by the system. The system successfully detected and captured 2 restricted activities as configured by the administrator.

- **User 2**: This user performed a total of 70 activities, all of which were non-restricted. As a result, the system did not capture any activities.
- **User 3**: A total of 30 activities were completed by this user. Out of these, 27 were non-restricted and ignored by the system, while 3 restricted activities were successfully captured.
- **User 4**: This user performed 100 activities, with 99 being non-restricted activities that were ignored by the system. The system captured 1 restricted activity as configured by the administrator.

Based on the activities observed, we calculated the accuracy of our Intrusion Detection System as follows:

- **Accuracy of Captured Activities**: Accuracy=(Expected Captured ActivitiesActual C aptured Activities)×100=(66)×100=100%
- **Accuracy of Ignored Activities**: Accuracy=(Expected Ignored ActivitiesActual Ign ored Activities)×100=(344344)×100=100%

CONCLUSION

In addition to our custom technologies, we have developed an intrusion prevention and alert system. Multiple modules are in place to monitor and log each user's activity within the system. Every action performed by a user is tracked and stored in a log file. If the system detects any abnormal behaviour—specifically, if a user attempts an action they are restricted from—the system will immediately alert the administrator. The system's self-monitoring feature ensures continuous observation of user activities.

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 indus trial 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 Techn

[8]. M.Nuaimi,L.C.Fourati,andB.B.Hamed,"Intellige ntapproachestoward in trusion 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.