Data Mining and Forensic Techniques for Internal Intrusion Detection and Protection System

Ms. Poonam Jadhavar[1] Prof. Varsha Dange[2]
[1]ME Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]Dhole Patil College of Engineering, Savitribai Phule University, Wagholi, Pune, India

Abstract— The IDPS uses a local computational grid to detect malicious behaviors in a real-time manner. In this paper, the system proposes a security system, named the Intrusion Detection and Protection System (IDPS for short) at system call level, which creates personal profiles for users to keep track of their usage habits as the forensic features. The proposed work is regarded with Digital forensics technique and intrusion detection mechanism. The number of hacking and intrusion incidents is increasing alarmingly each year as new technology rolls out. In this project, the system designed Intrusion Detection System (IDS) that implements predefined algorithms for identifying the attacks over a network. Therefore, in this project, a security system, named the Internal Intrusion Detection and Protection System (IIDPS), is proposed to detect insider attacks at SC level by using data mining and forensic techniques. The system can identify a user's forensic features by analyzing the corresponding SCs to enhance the accuracy of attack detection, and able to port the IIDPS to enhance the time efficiency we are using a Hadoop for process big data and K-Means theory for classification and a parallel system to further shorten its detection response time.

Key words: Data Mining, Insider Attack, Intrusion Detection and Protection, System Call (SC), Users' Behaviors

I. Introduction

As network-based computer systems have important roles in modern society, they have become the targets of intruders. Therefore, we need to find the best possible ways to protect our systems. The security of a computer system is compromised when an intrusion takes place. An intrusion can be defined as any action done to hamper the integrity, confidentiality or availability of the system. There are some intrusion prevention techniques which can be used to protect computer systems as a first line of defense. But only intrusion prevention is not enough. As systems become more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various penetration techniques. Therefore Intrusion detection is required as another measure to protect our computer systems from such type of vulnerabilities.

Traditional intrusion prevention techniques, such as firewalls, access control and encryption, have failed to fully protect networks and systems from increasingly sophisticated attacks and malwares. As a result, intrusion detection systems (IDS) have become an indispensable component of security infrastructure used to detect these threats before they inflict widespread damage. When building an IDS one needs to consider many issues, such as data collection, data pre-processing, intrusion recognition, reporting, and response. Among them, intrusion recognition is at the heart. Audit data are examined and compared with detection models, which describe the patterns of intrusive or benign behavior, so that both successful and unsuccessful intrusion attempts can be identified.

Artificial intelligence and machine learning techniques were used to discover the underlying models from a set of training data. Commonly used methods were rule based induction, classification and data clustering. In fact, the process of automatically constructing models from data is not trivial, especially for intrusion detection problems. This is because intrusion detection faces such problems as huge network traffic volumes, highly imbalanced attack class distribution, the difficulty to realize decision boundaries between normal and abnormal behavior, and requiring continuous adaptation to a constantly changing environment. Artificial intelligence and machine learning have shown limitations to achieving high detection accuracy and fast processing times when confronted with these requirements.

In the past decades, computer systems have been widely employed to provide users with easier and more convenient lives. However, when people exploit powerful capabilities and processing power of computer systems, security has been one of the serious problems in the computer domain since attackers very usually try to penetrate computer systems and behave maliciously, e.g., stealing critical data of a company, making the systems out of work or even destroying the systems. Generally, among all well-known attacks such as pharming attack, distributed denial-of-service (DDoS), eavesdropping attack, and spear-phishing attack, insider attack is one of the most difficult ones to be detected because firewalls and intrusion detection systems (IDSs) usually defend against outside attacks.

To authenticate users, currently, most systems check user ID and password as a login pattern. However, attackers may install Trojans to pilfer victims' login patterns or issue a large scale of trials with the assistance of a dictionary to acquire users' passwords. When successful, they may then log in to the system, access users' private files, or modify or destroy system settings. Fortunately, most current host-based security systems and network-based IDSs can discover a known intrusion in a real-time manner. However, it is very difficult to identify who the attacker is because attack packets are often issued with forged IPs or attackers may enter a system with valid login patterns. Although OS-level system calls (SCs) are much more helpful in detecting attackers and identifying users , processing a large volume of SCs, mining malicious behaviors from them, and identifying possible attackers for an intrusion are still engineering challenges.

Therefore, in this paper, The system propose a security system, named Internal Intrusion Detection and Protection System (IIDPS), which detects malicious behaviors launched toward a system at SC level. The IIDPS uses data mining and forensic profiling techniques to mine system call patterns (SC-patterns) defined as the longest system call sequence (SC-sequence) that has repeatedly appeared several times in a user's log file for the user. The user's forensic features, defined as an SC-pattern frequently appearing in a user's submitted SC-sequences but rarely being used by other users, are retrieved from the user's computer usage history. The contributions of this paper are identify a user's forensic features by analyzing the corresponding SCs to enhance the accuracy of attack detection; able to port the IIDPS to a parallel system to further shorten its detection response time; and effectively resist insider attack.

An intrusion detection system dynamically monitors the events taking place in a monitored system, and decides whether these events are symptomatic of an attack or constitute a legitimate use of the system. Depicts the organization of an IDS where solid arrows indicate data/control flow while dotted arrows indicate a response to intrusive activities. In general, IDSs fall into two categories according to the detection methods they employ, namely (i) misuse detection and (ii) anomaly detection. Misuse detection identifies intrusions by matching observed data with pre-defined descriptions of intrusive behavior. So well-known intrusions can be detected efficiently with a very low false positive rate. For this reason, the approach is widely adopted in the majority of commercial systems.

However, intrusions are usually polymorph, and evolve continuously. Misuse detection will fail easily when facing unknown intrusions. One way to address this problem is to regularly update the knowledge base, either manually which is time consuming and laborious, or automatically with the help of supervised learning algorithms. Unfortunately, datasets for this purpose are usually expensive to prepare, as they require labeling of each instance in the dataset as normal or a type of intrusion. Another way to address this problem is to follow the anomaly detection model proposed by Denning.

Anomaly detection is orthogonal to misuse detection. It hypothesizes that abnormal behavior is rare and different from normal behavior. Hence, it builds models for normal behavior and detects anomaly in observed data by noticing deviations from these models. There are two types of anomaly detection. The first is static anomaly detection, which assumes that the behavior of monitored targets never changes, such as system call sequences of an Apache service; the second type is dynamic anomaly detection. It extracts patterns from behavior habits of end users or networks/hosts usage history. Sometimes these patterns are called profiles.

Clearly, anomaly detection has the capacity of detecting new types of intrusions, and only requires normal data when building the profiles. However, its major difficulty lies in discovering boundaries between normal and abnormal behavior, due to the deficiency of abnormal samples in the training phase. Another difficulty is to adapt to constantly changing normal behavior, especially for dynamic anomaly detection.

II. Related work

In [1], Supervisory Control and Data Acquisition (SCADA) systems, which are widely used in monitoring and controlling critical infrastructure sectors, are highly vulnerable to cyber-attacks. Current security solutions can protect SCADA systems from known cyber assaults, but most solutions require human intervention. This paper applies autonomic computing technology to monitor SCADA system performance, and proactively estimate upcoming attacks for a given system model of a physical infrastructure. We also present the feasibility of intrusion detection systems for known and unknown attack detection. A dynamic intrusion response system is designed to evaluate recommended responses, and appropriate responses are executed to influence attack impacts. We used a case study of a water storage tank to develop an attack that modifies Modbus messages transmitted between slaves and masters. Experimental results show that, with little or no human intervention, the proposed approach enhances the security of the SCADA system, reduces protection time delays, and maintains water storage tank performance. In this paper, autonomic computing technology has been used to self-protect the SCADA system from cyber-attacks. This new technology integrates current security solutions so that the system can proactively monitor, estimate, detect, and react to known and unknown attacks with little or no human intervention. It also ensures the SCADA system is accessible 24/7. We applied the proposed approach to enhance the security of a SCADA system, which controls and monitors a water storage tank. Through the experimental result, we validated that the autonomic SCADA system maintained normal infrastructure operations and regulated the water level back to the normal operation region when alarm conditions were changed by attackers. The overhead time for identifying and protecting the SCADA system was short. It cost 22 sample time to regulate the water level back to normal. In the future, we will simulate more sophisticated cyber-attacks to validate the efficiency of the approach. In addition, we will also employ autonomic computing to self-protect the next generation SCADA systems from cyber assaults.

In[2], Intrusion detection based up on computational intelligence is currently attracting considerable interest from the research community. Characteristics of computational intelligence (CI) systems, such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information fit the requirements of building a good intrusion detection model. Here we want to provide an overview of the research progress in applying CI methods to the problem of intrusion detection. The scope of this review will be on core methods of CI, including artificial neural networks, fuzzy systems, evolutionary computation, artificial immune systems, swarm intelligence, and soft computing. The research contributions in each field are systematically summarized and compared, allowing us to clearly define existing research challenges, and to highlight promising new research directions. The findings of this review should provide useful insights into the current IDS literature and be a good source for anyone who is interested in the application of CI approaches to IDSs or related fi el ds.

In[3], A common application of virtual machines (VM) is to use and then throw away, basically treating a VM like a completely isolated and disposable entity. The disadvantage of this approach is that if there is no malicious activity, the user has to re-do all of the work in her actual workspace since there is no easy way to commit (i.e., merge) only the benign updates within the VM back to the host environment. In this work, we develop a VM commitment system called Secom to automatically eliminate malicious state changes when merging the contents of an OS-level VM to the host. Secom consists of three steps: grouping state changes into clusters, distinguishing between benign and malicious clusters, and committing benign clusters.

In[4], Autonomously detecting and recovering from faults is one approach for reducing the operational complexity and costs associated with managing computing environments. We present a novel methodology for autonomously generating investigation leads that help identify systems faults, and extends our previous work in this area by leveraging Restricted Boltzmann Machines (RBMs) and contrastive divergence learning to analyse changes in historical feature data. This allows us to heuristically identify the root cause of a fault, and demonstrate an improvement to the state of the art by showing feature data can be predicted heuristically beyond a single instance to include entire sequences of information.

The operational costs of large-scale computing environments are continuing to increase. In order to address this problem, self-managing systems are being developed that reduce the supervisory needs of computing environments. Self-healing systems are one such example, and operate by autonomously detecting then recovering from faults. Although there have been numerous advances in both of these aspects, most self-healing systems continue to require periodic human oversight. This constraint poses challenges for the continued reduction of costs, and restricts self-healing recovery strategies to reactive approaches.

In[5], Smart grid replaces analog mechanical meters with digital meters that record usage in real state of affairs. The power grid has been converted into an essential in the recent world. A smart grid is the integration of information and communications technology into electric transmission and distribution networks. Nowadays, the electricity make available manufacturing is grappling with an exceptional array of issues, period from a one requirement gap to getting higher expenses. In addition these and more forces are motivating the necessitate to pertaining the trade. Hence, it makes, is motivating the necessitate for a smart grid. With the increase in unit cost of electricity, there is a need for utilities to replace and renew aging transmission and distribution infrastructure with a pressure of using the assets wisely. Human errors and deliberate errors can be lowered by using smart instruments like smart meters. Smart grid can improve outage management performance by responding faster to repair equipment before it fails unexpectedly. The smart grid can improve load factors and reduce system losses. We can integrate renewable energy projects into the grid.

III. PROPOSED ALGORITHM

In proposed system, the system propose a security system, named Internal Intrusion Detection and Protection System (IIDPS), which detects malicious behaviors launched toward a system at SC level. In the IIDPS, the SCs collected in the class-limited-SC list, as a key component of the SC monitor and filter, are the SCs prohibited to be used by different groups/classes of users in the underlying system. To verify the feasibility and accuracy of the IIDPS, three experiments were performed. The first defined the decisive rate threshold between the user profile established for u and each of other users' user profiles. The outcome extends the features, confirming that data mining and forensic techniques used for intrusion detection provide effective attack resistance. The system need to study the SCs generated and the SC-patterns produced by these commands so that the IIDPS can detect those malicious behaviors issued by them and then prevent the protected system from being attacked.

The scientific method of gathering and examining information about the past which is then used in a court of law. The word forensic comes from the Latin term forēnsis, meaning "of or before the forum." The history of the term originates from Roman times, during which a criminal charge meant presenting the case before a group of public individuals in the forum. Both the person accused of the crime and the accuser would give speeches based on their sides of the story. The case would be decided in favor of the individual with the best argument and delivery. This origin is the source of the two modern usages of the word forensic – as a form of legal evidence and as a category of public presentation. In modern use, the term forensics in the place of forensic science can be considered correct, as the term forensic is effectively a synonym for legal or related to courts. However, the term is now so closely associated with the scientific field that many dictionaries include the meaning that equates the word forensics with forensic science.

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), an interdisciplinary subfield of computer science is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection), and dependencies (association rule mining). This usually involves using database techniques such as spatial indices. These patterns can then be seen as a kind of summary of the input data, and may be used in further analysis or, for example, in machine learning and predictive analytics. For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system.

Neither the data collection, data preparation, nor result interpretation and reporting are part of the data mining step, but do belong to the overall KDD process as additional steps.The contributions of this paper are identify a user's forensic features by analyzing the corresponding SCs to enhance the accuracy of attack detection able to port the IIDPS to a parallel system to further shorten its detection response time and effectively resist insider attack.

A. Advantages

– It identify a user's forensic features by analyzing the corresponding SCs to enhance the accuracy of attack detection

- It able to port the IIDPS to a parallel system to further shorten its detection response time.
- It effectively resist insider attack.
- The IIDPS can detect those malicious behaviors issued by them and then prevent the protected system from being attacked.
- The mining user profiles by using an unsupervised cluster approach can also improve the performance of the mining process.

B. *System Architecture*

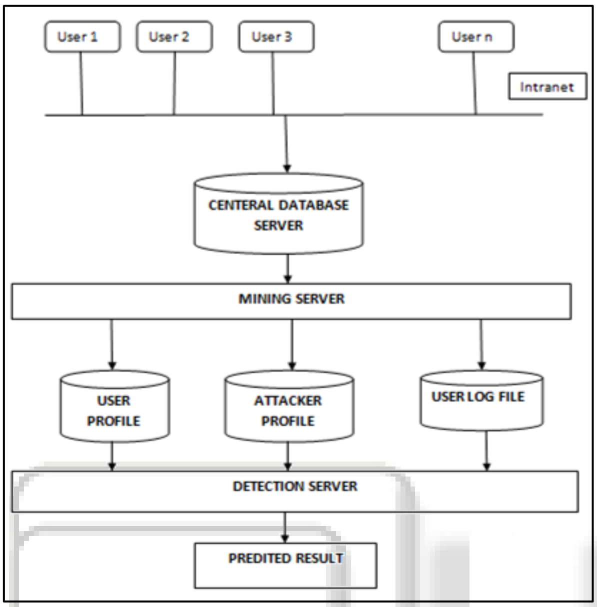


Fig. 1: System Architecture

C. *Modules Description*

1) *User Habits*

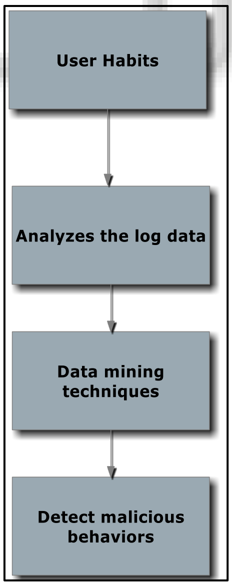


Fig. 2: Use Habits

The mining server analyzes the log data with data mining techniques to identify the user's computer usage habits as his/her behavior patterns, which are then recorded in the user's user profile. The detection server compares users' behavior patterns with those SC-patterns collected in the attacker profile, called attack patterns, and those in user profiles to respectively detect malicious behaviors and identify who the attacker is in real time. When an intrusion is discovered, the detection server notifies the SC monitor and filter to isolate the user from the protected system. The purpose is to prevent him/her from continuously attacking the system.

2) *Mining Server*

Both the detection server and the mining server are run on the local computational grid to accelerate the IIDPS's online detection and mining speeds and enhance its detection and mining capability. If a user logs in to the system by using another person's login pattern, the IIDPS identifies who the underlying user is by computing the similarity scores between the user's current inputs, i.e., SCs, and the behavior patterns stored in different users' user profiles. In the IIDPS, the SCs collected in the class-limited-SC list, as a key component of the SC monitor and filter, are the SCs prohibited to be used by different groups/classes of users in the underlying system.

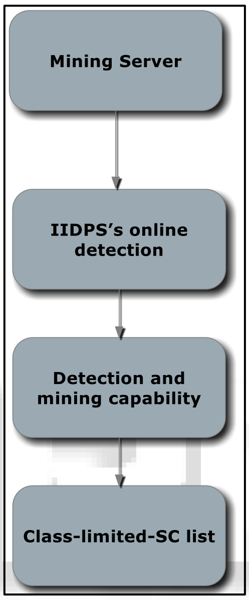


Fig. 3: Mining server

3) *Detection Server*

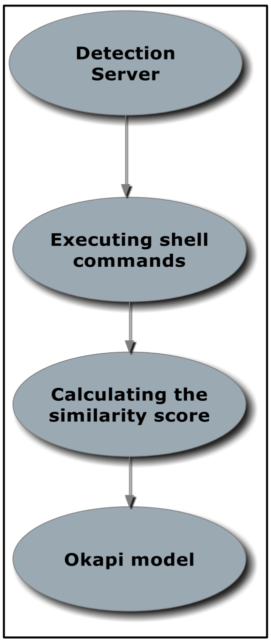


Fig. 4: Detection server

The detection server captures the SCs sent to the kernel by the underlying user u when u is executing shell commands and stores the SCs in the u's log file. After this, the server tries to identify whether u is the underlying account holder or not by calculating the similarity score between the newly generated SCs, denoted by NSCu, in the u 's current inputs and the usage habits, i.e., forensic signatures, stored in u 's user profile to verify u . The Okapi model, which is utilized to calculate the similarity score between user j 's user profile Uhj and an unknown user u 's current input SC-sequence, denoted bySim (u,j) , is defined as

$$Sim(u,j) = \sum_{i=1}^{p} F_{iu} \cdot W_{ij}$$

4) *Attackers*

An attack pattern, which may be an attacker-specific pattern or a pattern commonly used by attackers, can be identified in the same method. Similarly, an attack pattern that an attacker frequently submits but others have seldom or never submitted will be considered as one of the attacker's representative attack patterns and will obtain a high similarity weight. Hence, signatures collected in an attacker profile can be classified into common signatures and attacker-specific signatures. The latter can be used to identify who the possible attackers are when a protected system is attacked by attacker-specific signatures.

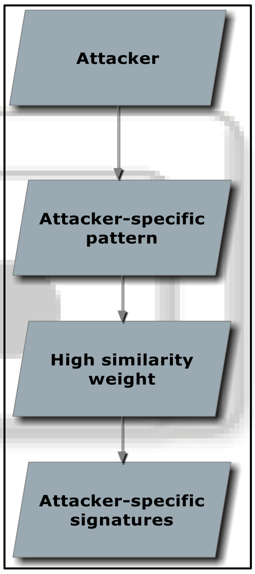


Fig. 5: Attacker

5) *Performance Analysis*

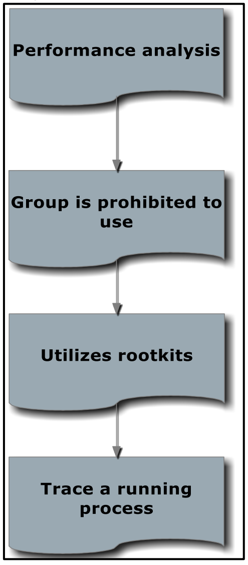
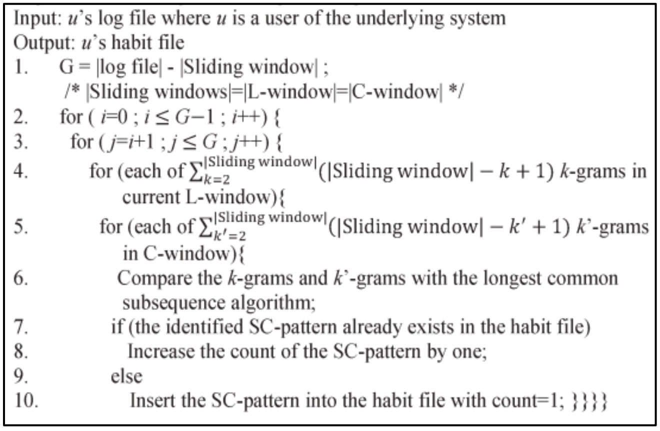


Fig. 6: Performance Analysis

An insider attacker may log in to a system by using another user's login ID and password and do something maliciously. As previously stated, a Type-I attack is an attack in which a certain group is prohibited to use. A Type-II attack utilizes rootkits to issue sensitive SCs, i.e., unlinkat() and kill(), to modify sensitive data or resource of a system. A Type-III attack employs GDB, i.e., a program debugging tool, to trace a running process and launches a DDoS attack to intrude an outside-world system.

D. *Algorithm Description*

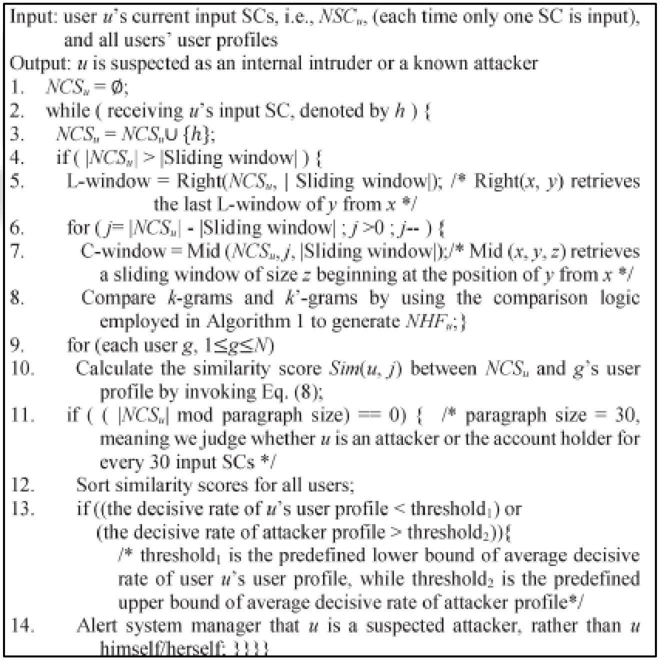
1) *Algorithm for generating a user habit file:*



```
Input: $u$'s log file where $u$ is a user of the underlying system
Output: $u$'s habit file
G = |log file| - |Sliding window| ;
  /* |Sliding windows|=|L-window|=|C-window| */
for ( $i$=0 ; $i \le G-1$ ; $i$++) {
  for ( $j$=$i$+1 ; $j \le G$ ; $j$++) {
    for (each of $\sum_{k=2}^{|\text{Sliding window}|}(|\text{Sliding window}| - k + 1)$ $k$-grams in
        current L-window){
      for (each of $\sum_{k'=2}^{|\text{Sliding window}|}(|\text{Sliding window}| - k' + 1)$ $k'$-grams
          in C-window){
        Compare the $k$-grams and $k'$-grams with the longest common
          subsequence algorithm;
        if (the identified SC-pattern already exists in the habit file)
          Increase the count of the SC-pattern by one;
        else
          Insert the SC-pattern into the habit file with count=1; }}}}
```

Fig. 7: Algorithm for generating a user habit file

2) *Detecting an internal intruder or an attacker*



```
Input: user $u$'s current input SCs, i.e., $NSC_u$, (each time only one SC is input),
       and all users' user profiles
Output: $u$ is suspected as an internal intruder or a known attacker
$NCS_u = \emptyset$;
while ( receiving $u$'s input SC, denoted by $h$ ) {
  $NCS_u = NCS_u \cup \{h\}$;
  if ( $|NCS_u|$ > |Sliding window| ) {
    L-window = Right($NCS_u$, | Sliding window|); /* Right($x$, $y$) retrieves
                 the last L-window of $y$ from $x$ */
    for ( $j$= $|NCS_u|$ - |Sliding window| ; $j$ >0 ; $j$-- ) {
      C-window = Mid ($NCS_u$, $j$, |Sliding window|);/* Mid ($x$, $y$, $z$) retrieves
        a sliding window of size $z$ beginning at the position of $y$ from $x$ */
      Compare $k$-grams and $k'$-grams by using the comparison logic
        employed in Algorithm 1 to generate $NHF_u$;}
    for (each user $g$, $1 \le g \le N$)
      Calculate the similarity score $Sim(u, j)$ between $NCS_u$ and $g$'s user
        profile by invoking Eq. (8);
    if ( ( $|NCS_u|$ mod paragraph size) == 0) {   /* paragraph size = 30,
        meaning we judge whether $u$ is an attacker or the account holder for
        every 30 input SCs */
      Sort similarity scores for all users;
      if ((the decisive rate of $u$'s user profile < threshold$_1$) or
         (the decisive rate of attacker profile > threshold$_2$)){
           /* threshold$_1$ is the predefined lower bound of average decisive
           rate of user $u$'s user profile, while threshold$_2$ is the predefined
           upper bound of average decisive rate of attacker profile*/
      Alert system manager that $u$ is a suspected attacker, rather than $u$
        himself/herself; }}}}
```

Fig. 8: Detecting an internal intruder or an attacker

IV. CONCLUSION AND FUTURE WORK

In this paper, an IIDPS is developed to detect insider attacks at SC level by using data mining and forensic techniques. The experimental results show that the IIDPS can effectively resist several aforementioned attacks. This process confirming that data mining and forensic techniques used for intrusion detection provide effective attack resistance and also shows IIDPS may detect inaccurately when user's habit suddenly changes. Nevertheless, in most cases, the IIDPS can still identify the legality of a login user. When a user inputs a command, hundreds or thousands of SCs will be generated. Analyzing a huge number of SCs often takes a long time. The IIDPS spends 0.45 s to identify a user. Although other systems consume longer time for data analysis than the IIDPS does, how to mine SCs in an efficient method should be addressed. Employing a local computational grid can accelerate the processing speed of the miming server and detection server. A mathematical analysis on the IIDPS's behaviors is helpful in deriving its formal performance and cost models, with which users can predict performance and cost of the IIDPS before using it. This can also detect malicious behaviors for systems employing GUI interfaces and then prevent the protected system from being attacked. The proposed model can be further used to increase detection accuracy and improve the decisive rate.

REFERENCES

[1] S. Gajek, A. Sadeghi, C. Stuble, and M. Winandy, "Compartmented se-curity for browsers—Or how to thwart a phisher with trusted computing," in Proc. IEEE Int. Conf. Avail., Rel. Security , Vienna, Austria, Apr. 2007, pp. 120–127.
[2] C. Yue and H. Wang, "BogusBiter: A transparent protection against phishing attacks," ACM Trans. Int. Technol., vol. 10, no. 2, pp. 1–31, May 2010.
[3] Q. Chen, S. Abdelwahed, and A. Erradi, "A model-based approach to self-protection in computing system," in Proc. ACM Cloud Autonomic Comput. Conf. , Miami, FL, USA, 2013, pp. 1–10.
[4] F. Y. Leu, M. C. Li, J. C. Lin, and C. T. Yang, "Detection workload in a dynamic grid-based intrusion detection environment," J. Parallel Distrib. Comput., vol. 68, no. 4, pp. 427–442, Apr. 2008.
[5] H. Lu, B. Zhao, X. Wang, and J. Su, "DiffSig: Resource differentiation based malware behavioral concise signature generation," Inf. Commun. Technol., vol. 7804, pp. 271–284, 2013.
[6] Z. Shan, X. Wang, T. Chiueh, and X. Meng, "Safe side effects commit-ment for OS-level virtualization," inProc. ACM Int. Conf. Autonomic Comput., Karlsruhe, Germany, 2011, pp. 111–120.
[7] M. K. Rogers and K. Seigfried, "The future of computer forensics: A needs analysis survey," Comput. Security, vol. 23, no. 1, pp.12–16, Feb. 2004.
[8] J. Choi, C. Choi, B. Ko, D. Choi, and P. Kim, "Detecting web based DDoS attack using MapReduce operations in cloud computing environment," J. Internet Serv. Inf. Security, vol. 3, no. 3/4, pp. 28–37, Nov. 2013.
[9] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "MIS: Malicious nodes identification scheme in network-coding-based peer-to-peer streaming," in Proc. IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1–5.
[10] Jonathon T. Giffin, Somesh Jha, and Barton P. Miller "Automated Discovery of Mimicry Attacks", 2006.