I J C T A, 9(14) 2016, pp. 6377-6383
© International Science Press

Automated Discovery of Mimicry Attacks

Gaurang S. Joshi* **and R. Subash***

ABSTRACT

A security system, named the Detection and Protection System, is employed to detect insider attacks which are mimicked at SC (System call) level by using mining techniques. Analyzing system calls (SCs) generated by commands at the system level can identify these commands, with which attacks were detected accurately, and attack patterns are the features of attack. The IIDPS (Internal Intrusion Detection and Protection System) creates users' personal profiles to keep track of users' usage habits as their forensic features and determines whether a valid login user is the account holder or not by comparing user's current computer usage nature with the patterns collected in the account holder's personal profile.

Keywords: data mining, IIDPS, insider attack, SC (system call)

I. INTRODUCTION

Security has been a prime aspect to be taken care of in the computer domain. The most difficult attack to detect is insider attacks among pharming attack; distributed denial-of-service (DDoS), eavesdropping attack, and spear-phishing attack [1] because firewalls and intrusion detection systems (IDSs) usually defend against outside attacks. But the insider attacks is a greater threat where security is concerned [2]. Currently, most systems check user ID and password as a login pattern. However, attackers may install Trojans to manipulate victims' login patterns or issue a large scale of trials with the assistance of a dictionary to acquire users' password when successful, they may then login to the system, access users' private files, or modify or destroy system settings. Most current host-based security systems and network-based IDSs can discover a known intrusion in a real-time manner.

Interruption in real time manner can be supervised by [3]. Currently, most host-based security systems [4] and network-based IDS [5] employ the same. Intrusion detection systems monitor network and system activities to prevent attacks and malicious activities that can come from within a network. Typically, intrusion detection system will notify the network administrator when suspicious activity is initiated. In some cases, the intrusion detection systems can take action when problems are detected such as barring a user or IP address from accessing the system. Since a large number of OS-level system calls were generated, mining malevolent behaviors from them and identifying possible intruders remains a major engineering challenge. Computer forensics science, based on a security event [6], view computer systems as crime scenes, identify, recover, analyze and present opinions the on respective basis. Attackers spread computer viruses, malicious codes and also conduct DDoS attacks [7].

Hence, the designed system, the Internal Intrusion Detection and Protection System (IIDPS) detects probable intruder behaviors launched at the system level. An intrusion detection system (IDS) is a device or software application that observers network or system activities for malicious activities or policy violations and produces electronic reports to a management station. The forensic features of the user, defined by the system call pattern they follow are recorded for references of the identity of the user were determined from the user's computing history. After identifying user's usage habits, the corresponding SCs are analyzed to

* Computer Science and Engineering, SRM University, Chennai, India, *E-mail: gaurang_sanjay@srmuniv.edu.in; subash.r@ktr.srmuniv.ac.in*

enhance the accuracy of attack detection. The IIDPS guarantees better average detection accuracy. The following sections describe the system framework and algorithms required for implementation.

II. IIDPS

(A) System Framework

The system structure consists of SC monitor and filter, as a loadable module in a kernel of the system which collects SCs submitted to the kernel and store them in the format of {u_id, p_id, SC}, where u_id is the user id, p_id is the process id and SC is the system call generated. Two algorithms are implemented to build a user habit file and to detect an internal intruder. The mining server analyzes the user log data with data mining techniques to identify user's computer usage habits and further recorded in the user's user profile. The Detection server compares user patterns with the SC patterns collected in attacker profile, and those in user profiles respectively detect malicious behaviors and identify the attacker in real time. Also, a local computational grid is required to store user log files, user profiles, and attacker profiles.

The detection server and the mining server are implemented on the local computational grid to enhance IIDPS's detection accuracy and removal capability.

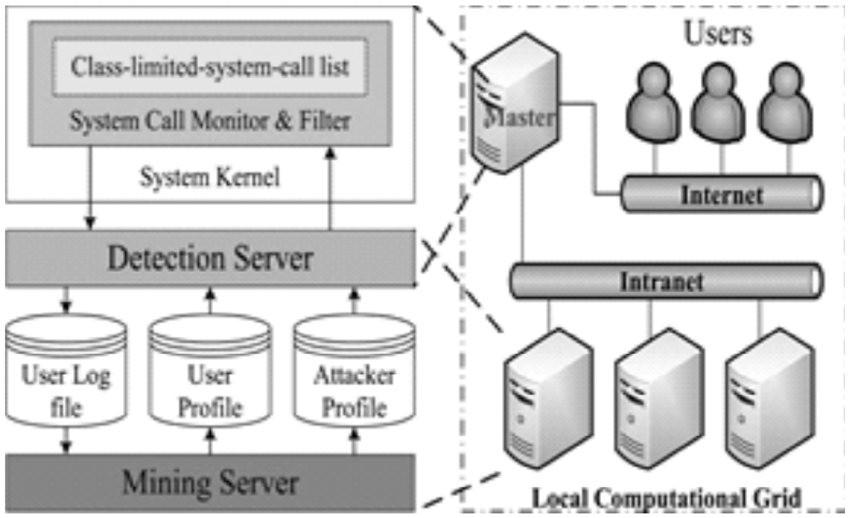

Figure 1: System framework. [15]

(B) SC Monitor and Filter

A System Call is an edge between a user application and services provided by the kernel. A lot of SCs was generated during the execution of a task. It is hard for a system to monitor all SCs at the same time, particularly when many users are running their programs. Therefore, we should filter out some commonly used safe SCs that do not affect the profiles to handle this problem, the statistical model of term frequency-inverse document frequency (TF-IDF) is used to sort the required SCs collected in the user log file.

Term Frequency is represented by,

$$TFi,\, j = {n(i,\, j)}\Big/_{k=1} n(k,\, j) \tag{1}$$

Where n(i,j) is the number of times that ti was issued during the execution of j, h is the number of different SCs generated when j is executed, and the denominator k=h n(k,j) sums up the numbers of times that all these SCs were launched. The inverse document frequency (IDF), the measure of the importance of ti among all concerned shell commands, is defined as,

$$IDFi = \log|D| \big/ |\{j : t \in dj\}| \tag{2}$$

Here |D|, the cardinality of D, is the number of shell commands in the dedicated corpus and {j: ti " dj} is the set of shell commands dj. In which each member employs 'ti' during its execution. The TF-IDF weight of ti generated by j is defined as

$$(TF - IDF)i, j = TFi, j \times IDFi \tag{3}$$

Below are examples of commands and the number of system calls generated respectively.

The TF weight of close () in the command chmod from (1) is 0.2021(=19/94) where 94 is the total number of times that SCs were generated from 'chmod' command. A data mining tool named, iDataAnalyzer [8], is used to calculate class predictability and class predictiveness. These parameters are used to evaluate intraclass and interclass weights.

Class predictability: Given a class *C* and a categorical attribute *A* with v_1, v_2, v_3, . . . , and v_n as its candidate values, class C's predictability score on A=vi is defined as the percentage with which A's value in C is vi.

Class predictiveness: Given a class C and its categorical attribute with v1, v2, v3,…and vn as candidate values, the attribute value vi's predictiveness score P (A=vi) is the probability of which T with A=vi resides in C.

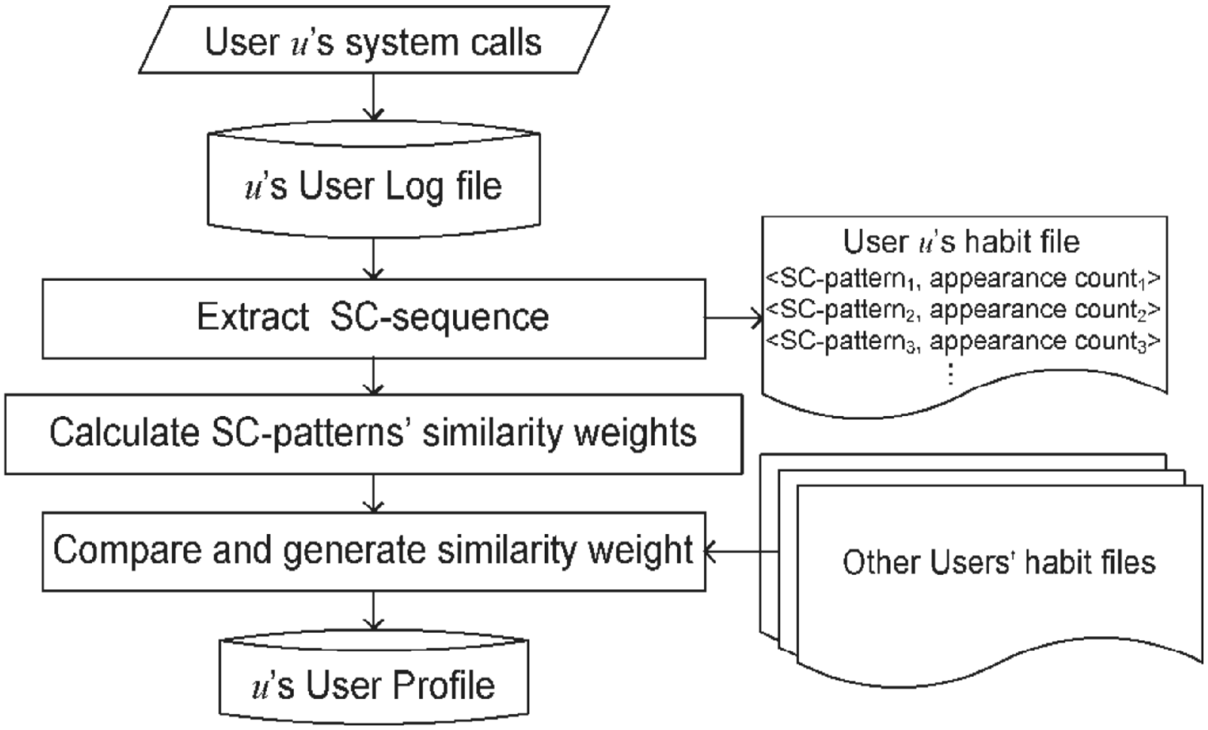
User u's system calls
u's User Log file
Extract SC-sequence
User u's habit file
<SC-pattern1, appearance count1>
<SC-pattern2, appearance count2>
<SC-pattern3, appearance count3>
Calculate SC-patterns' similarity weights
Compare and generate similarity weight
Other Users' habit files
u's User Profile

Figure 2: Generation of the user profile [15]

(C) Mining Server

The mining server extracts SC-sequence generated by the user's log file. The resultant SC pattern appears in the file, and the results were stored in the pattern of {SC pattern, appearance counts} in user's habit file. The similarity weights of SC patterns are matched to remove commonly used SC patterns. Further, the result is matched with other users' habit wallet to identify user's exact behavior in the SC-patterns.

1. Mining User and Attacker Habits

The SCs which were collected in user's log file were processed by the IIDPS with a sliding window, named as log –sliding window (L-window). The SCs were partitioned in the window of l-grams where l is the number of consecutive SCs. Along with it, another window of the same size is determined called compared-sliding window (C-window) is used to identify other patterns in the user's log file. This time, l' consecutive SCs are extracted from the C-window to generate the total of l'=2, 3, 4…|sliding window|.

The mining server compares l-grams and l'-grams. All the l grams from the L-window and all the l' grams from C-window were compared. The longest common subsequence algorithm was used for the comparing of above. The algorithm for generating user's habit file is as follows.

```
Input: u's log file where u is a user of the underlying system
Output: u's habit file.
1. G = |log file| - |sliding window|
/*|sliding windows|=|L-window|=|C-window|/*
2. for(i = 0; i <= G-1; i ++){
3. for(j=i+1; j <= G; j++){
4. for (each of $\Sigma_{k=2}^{|sliding window|}$(|sliding window| - k+1) k-grams in current L-window){
5. for (each of "$_{k'=2}^{|sliding window|}$(|sliding window| - k'+1) k'-grams in C-window){
6. Compare the k-grams and k'-grams with the longest Common Subsequence algorithm.
7. if (the identified SC-pattern already exists in the habit file)
8. Increase the count of the SC-pattern by one;
9. else
10. Insert the SC-pattern into habit file with count = 1;}}}}
```

Figure 3: Algorithm to generate user's habit file [15]

2. Creating User Profiles and Attacker Profiles- As seen in Figure 2, a user's user profile is a habit file with an SC-appearance count. It was substituted by its corresponding similarity weight.

Those SC patterns with fewer similarity weights were removed since they are not useful to differentiate between the other users.

Similarly, an attack pattern which may be specific to attackers is identified in the same way. Also, an attack pattern that has been submitted by the attacker but never submitted by others will obtain a high similarity weight.

A commonly used equation (4), is used to assign weight to a term in information retrieval domain [], is used to give each SC pattern a similarity weight.

$$Wij = \frac{fij \ \log_{Mj} N + 0.5}{fij + 0.5 + 1.5 \times nsj \times \log(N+1)} \tag{4}$$

i = 1, 2, 3… k and J= 1, 2, 3… N

In which fij is the appearance count of CSi in UHj , nsj is the total number of SC-patterns collected in UHj, nsavg is the average number of SC-patterns that an element of D has, and log ((N + 0.5)/Mi)/ log (N + 1) is the inverse characteristic profile frequency (ICPF) [9].

(D) Detection Server

The Detection Server captures the sent SCs by the user to the kernel, when users are executing shell commands and were stored in user's log file. After this, the server tries to check whether the user is underlying account holder or not by comparing similarity scores between newly generated SCs and the user's usage habits. The Okapi model [10] is used to calculate a similarity score between user's profile and an unknown user u's current input SC-sequence, denoted by Sim(u, j).

$$\sum_{i=(5)}^{p} (u, j) = \sum Fiu, Wij$$

In which p is the number of SC-patterns appearing in both NCSu and UHj , Fiu is the appearance count of SC patterns collected in NCSu, and Wij produced by invoking is the similarity weight of i in UHj .The higher the Sim(u, j), the higher the probability, with which u is the person j submitted NCSu. The concept of the Detection Server is same as the Mining server only difference being that the comparison between L-window and C-window is from the back to front each time when an SC is an input by the user. That is, when the L-window contains the last |sliding window| SCs, the C-window left shifts one SC from L-window and compares l grams and l' grams. After this process, for each left shift on C-window covers the first |sliding window| SCs of NCSu.

```
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's is suspected as an internal intruder
1. $NCS_u$ = Φ;
2. while( receiving u's input SC, denoted by h) {
3. $NCS_u$ = $NCS_{u\ ä}$ {h};
4. if (| $NCS_u$| > |sliding window|) {
5. L-window = Right($NCS_u$; |sliding window|; /*Right (x,y) retrieves the last L-window of y from x*/
6. for(j=|$NCS_u$| - |sliding window|; j>0; j—) {
7. 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 */
8. Compare k-grams and k'grams by using comparison logic employed in Algorithm 1 to generate $NHF_u$}
9. for (each user g, 1<=g<=N)
10. Calculate the similarity score Sim(u, j) between $NCS_u$ and g's user profile by invoking equation (5).
11. 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*/
12. Sort similarity scores for all users;
13. if (((the decisive rate of u's user profile < threshold;) or (the decisive rate of attacker profile > threshold;)){ /* threshold; is the predefined lower bound of average decisive rate of user u's profile, while threshold; is predefined upper bound of average decisive rate of attacker profile */
14. Alert system manager that u is a suspected attacker, rather than u himself ;}}}}
```

Figure 4: Detection server detects whether the user is a possible intruder.

When the Detection Server algorithm was invoked, the sliding window left shifts, to identify the SC sequences on each new input of SC. If the user's input are less than or equal to |sliding window|, no action is performed.

(E) Computational Grid

The computational grid is a collection of internally connected computers working together as a single integrated computing resource. A Mining server and a Detection server have been implemented in this paper to speed up data processing. The advantage of cluster computing is that it divides a large task into smaller subtasks, supported by multiple computer processors. The master node coordinates the distributed processing and combines processing results of all processors to complete the entire computation.

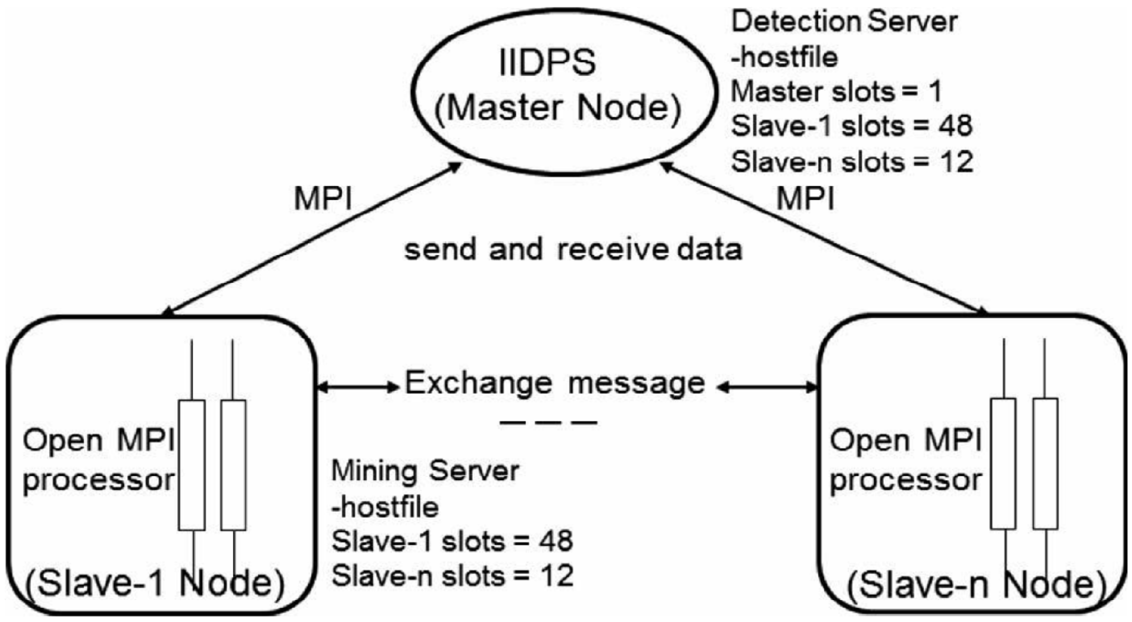

Figure 5: Symmetric multiprocessing cluster (SMP) with open message passing interface (MPI)[15]

The functions of Mining server were fulfilled by grid processors with a hostfile that defines computation resources for mining server. The Detection server is implemented on the master node of the computational grid with hostfile describing computation resources for the detection server.

III. SUMMARY

An IIDPS is developed to detect and resist insider attacks at SC level by using data mining and forensic techniques, extending the features of [11]. A command input by a user will generate hundreds and thousands of SCs. The IIDPS, on an average, takes 0.45s to identify a user. Implementing a local computational grid can increase the processing speed of the mining and the detection server. Systems employing GUI interfaces can also detect malicious behaviors despite user behavior profiles. Many third party shell commands [12], [13] have been developed, including those which are present in Oracle Database, Oracle Web Logic, and IBM WebSphere MQ. Detecting attacker signatures [14] has also being implemented for further accuracy.

IV. CONCLUSION

In this paper, IIDPS has been proposed an approach that employs Data Mining and Forensic Techniques to identify the representative SC-patterns for a user.

The time that a habitual SC pattern appears in the user's log file would be counted, the most commonly used SC-patterns are filtered out, and then a user's profile is established. The further study will be done by improving IIDPS's performance and investigating third-party shell commands.

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, 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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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

[8] R. J. Roger and M. W. Geatz, *Data Mining: A Tutorial-Based Primer*. Reading, MA, USA: Addison-Wesley, 2002.

[9] D. Zhu and J. Xiao, “R-tfidf, a Variety of tf-idf Term Weighting Strategy in Document Categorization,” in *Proc. Int. Conf. Semantics, Knowledge Grids*, Beijing, China, Oct. 2011, pp. 83–90.

[10] S. E. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, and A. Payne, “Okapi at TREC-4,” in *Proc. 4th text Retrieval Conf.*, 1996, pp. 73–96.

[11] F. Y. Leu, K. W. Hu, and F. C. Jiang “Intrusion detection and identification system using data mining and forensic techniques,” *Adv. Inf. Comput. Security*, vol. 4752, pp. 137–152, 2007.

[12] Symantec CSP, “Executing operating system commands from PL/SQL,” Oracle, Redwood City, CA, USA, White Paper, Jul. 2008.

[13] J. Böhm-Mäder, “The WebSphere MQ for UNIX administration tool,” Free Softw. Found., Boston, MA, USA, May 2013.

[14] S. O’Shaughnessy and G. Gray, “Development and evaluation of a data set generator tool for generating synthetic log files containing computer attack signatures,” Int. J. Ambient Comput. Intell., vol. 3, no. 2, pp. 64–76,Apr. 2011

[15] Fang-Yie Leu, Kun-Lin Tsai, *Member, IEEE*, Yi-Ting Hsiao, and Chao-Tung Yang “An Internal Intrusion Detection and Protection System by Using Data Mining and Forensic Techniques”.