AUTOMATIC INTERNAL FORENSIC MECHANISM FOR INTRUSION DETECTION

Ms. Dhonde Pooja Panditrao[1], Ms. Jadhav Ashwini Rustumrao,[2] Ms.Dake Pooja Dattatray,[3] Ms.Bansode Priyanka Vinayak,[4]

[1]*Ms.Dhonde Pooja Panditrao,Department of Computer Engineering,PREC loni, Maharashtra, India.*
[2]*Ms. Jadhav Ashwini Rustumrao, Department of Computer Engineering, PREC loni, Maharashtra, India.*
[3]*Ms.Dake Pooja Dattatray,Department of Computer Engineering,PREC loni, Maharashtra, India.*
[4]*Ms.Bansode Priyanka Vinayak, Department of Computer Engineering,PREC loni, Maharashtra, India.*

ABSTRACT

Now days, to authenticate users as the login patterns, most computer systems use user IDs and passwords. However, many people share their login patterns with co-workers and request these co-workers to assist co-tasks, thereby making the paradigm as one of the weakest points of computer safety. Insider attackers, the valid users of a system who attack the system internally, are difficult to invent since most intrusion detection systems and firewalls identify and isolate malicious behaviours exposed from the external world of the system only. In addition, some studies claimed that analysing system calls (SCs) generated by commands can recognize these commands, with which to accurately detect attacks, and attack patterns are the features of an invasion. so, in this paper, a security system, named the Internal Intrusion Detection , Protection System (IIDPS), are developed to invention insider attacks at SC level by using data mining and forensic techniques. The IIDPS synchronize users' personal profiles to keep path of users' usage habits as their forensic features and determines whether a authenticated login user is the account holder or not by comparing his/her current computer usage behaviours with the patterns gathered in the account holder's personal profile. The experiment all results analysed that the IIDPS's user identification precision is 94.29%, whereas the response time is less than 0.45 s, implying that it can servive a protected system from insider invasion impressive and efficiently.

1. INTRODUCTION:

Computer forensics science, which shows computer systems as crime scenes, aims to identify, preserve, recover, analyse, and present review on information collected for a security event [7]. It analyses what attackers have done such as spread viruses, malwares, and malicious codes and handled DDoS attacks [8]. Most intrusion detection techniques focus on how to find malicious network behaviours[9], [10] and acquire the characteristics of attack packets,i.e., attack patterns, based on the histories stored in log files [11], [12]. Qadeeret al. [13] used self-developed packet sniffer to collect network packets with which to distinguish network attacks with the help of network states and packet distribution. O' Shaughnessy and Gray [14] achieve network intrusion and attack patterns from system log files. These files contain used to detect computer misuse. It means that, from synthetically generated log files, these traces or patterns of misuse can be more accurately regenerate. Wu and Banzhaf [15]overviewed research progress of applying methods of computational smartness, including artificial neural networks, fuzzy systems, invented computation, artificial immune systems, and swarm intelligence, to find various malicious behaviours. The authors systematically summarized and compared different intrusion detection methods, thus permitted us to clearly view those existing research challenges.

Hence, in this paper, we propose a security system, named Internal Intrusion Detection and Protection System (IIDPS), which invention malicious behaviours launched toward a system at SC status. The IIDPS uses data mining and forensic profiling techniques to detect system call patterns (SC-patterns) defined as the longest system call precede (SC-sequence) that has continuously appeared many times in a user's log file for the user. These user's forensic features, defined as an SC-sample frequently appearing in a user's submitted SC-sequences but rarely being used by other users, are recover from the user's computer usage history pattern. The

contributions of this paper are: 1) identify a user's forensic features by analysing the corresponding SCs to enhance the accuracy of attack detection; 2) capable to port the IIDPS to a contrast system to future shorten its detection response time; and 3) effectively resist insider attack.

2. EXISTING SYSTEM:

Internal attackers, the valid users of a system who attack the system internally, are hard to detect since many intrusion detection systems and firewalls specify and isolate malicious behaviours launched from the outside world of the systemonly.

3. PROPOSED SYSTEM:

In this paper, we first introduce the IIDPS framework and describe components of the IIDPS in detail. Two algorithms are used to present for generating a user habit file and detecting an internal intruder also using web cam camera tacking attacker photo, and link he/she used that is send to host Email. It is helpful to find out attacker and if know the activity of attacker then it easy to recover or restore.

4. LITERATURE SURVEY:

Techniques Fang-YieLeu, 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" 1932-8184 © 2015

In this paper, we have 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 is counted, the most commonly used SC-patterns are filtered out, and then a user's profile is established. By identifying a user's SC-patterns as his/her computer usage habits from the user s current input SCs, the IIDPS resists suspected attackers. The administrators to point out an insider or an attacker in a closed environment. The further study will be done by improving IIDPS's performance and investigating third-party shell commands.

Chen, S. Abdelwahed, and A. Erradi, "A model-based approach to self-protection in computing system, in Proc. ACM Cloud Autonomic Compute Conf., Miami, FL, USA, 2013, pp. 110.

Approach to estimate, detect and identify security attacks along with planning a sequence of actions to effectively protect the networked computing system this paper introduces a model-based autonomic security management (ASM). Sensors collect system and network parameters and send the data to the fore- casters and the intrusion detection systems (IDSes) In the proposed approach to recover the system based on the signature of attacks A multi-objective controller selects the optimal protection method. On several case studies including Denial of Service (DoS) attacks, SQL Injection attacks and memory exhaustion attacks, the proposed approach is demonstrated. Experiments show that from known and unknown attacks the ASM approach can successfully defend and recover the victim host while maintaining QoS with low overheads.

5. USER PROFILES:

5.1 User Module:

In this system, user is one of actor which may be either authorized or unauthorized. Here user can modify the files, delete the files, or insert the files. User forensic profile is stored on host machine that can be used to detect the malicious activity of the user.

5.2 Client Module:

Client is the user of the system that can monitor the user activity and creates the log and capture the user image if he/she tries to access the restricted files. All process logs sends to the server for processing and recovered the files if modified. In this module, the deleted, inserted and updated file logs are sent to the server. If user performs any intrusion, then the system capture the user image and send it to the clients emailed. Also, it get backup of all deleted, updated and inserted files. We are using AES algorithm to encrypt the log of the client. The logs are encrypted when it send to the server. Using this we can achieve the security in our system.

5.3 Server Module:

Server checks the received process log from the client for integrity check. If server. In this module, Server maintains the history of client machine. It keeps record of all details of the client machine. Server send the intrusion log to the client emailed. Server sends the every log to the clients emailed. If user perform any intrusion, its log is maintained on the server and send it to the client emailed. When log is came from client, it is in encrypted form. At server side, this log is decrypted.

6. MATHEMATICAL MODULE:

Set (U) = u0, u1, u2, u3 U0-insert files
U1-delete files U2-update files

U3-install new process

Client:
Set (C) = c0, c1, c2, c3, c4, c5, c6 C0-capture user image

C1- generate file log
C2-generate process log
C3-encrypt all logs using AES algorithm C4- send encrypted files log to server
C5- send encrypted processes log to server
C6-send user image to client emailed

6.1 Server:
Set (S) = c4, c5, s0, s1, s2, s3 S0-decrypt all encrypted logs
S1-send files log to client email id
S2-send process log to client email id S3-maintain history of client's pc
Union and Intersection of project:-
Set (C) = c0, c1, c2, c3, c4, c5, c6 Set (S) = c4, c5, s0, s1, s2, s3
C union S= c0, c1, c2, c3, c4, c5, c6, s0, s1, s2, s3 C intersection S = c4, c5

7. VEIN DIAGRAM:

Following figure shows the interaction of the project module with each other .we have shown this interaction by using set theory.

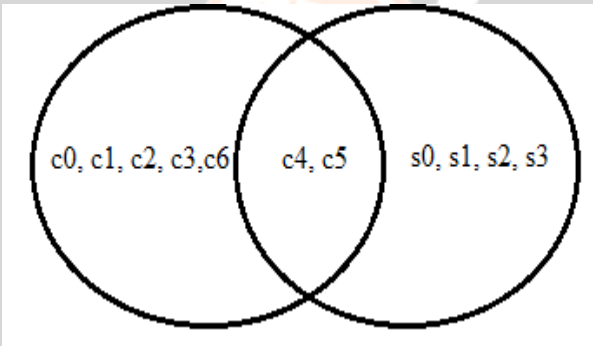

Figure1 : Intersection module R and Set B

8. FLOW OF SYSTEM:

This system can be used to detect the host intrusion detection where host machine comprises the confidential files. Attackers can attack on host machine that attacks would be detected by the system and updated files can be recovered by system. This system can detect the files modification and also prevent the file modification.

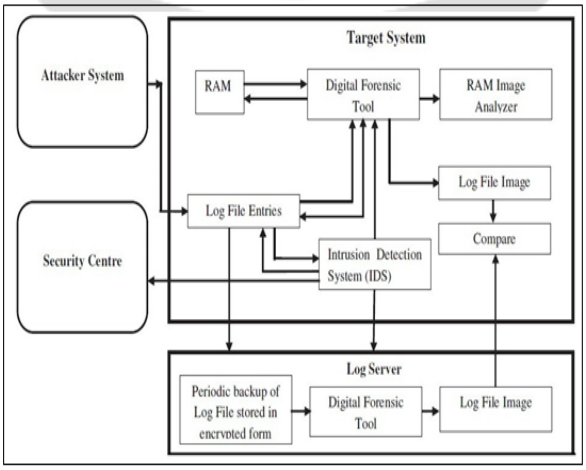

Figure2- Flow of System

9. ALGORITHMS:

Input: U's log file where U is user of the host machine.
Output: U's habit file or Attack Detection.
Procedure:

```
        G = |Log File|-|Sliding  Window|
        |Sliding  Window| = |L-Window|  = |C-Window|
for(i = 0; i < G-1;  i++)
        {
for( j = 0; j < G-1;  j++)
         {
add K grams of L window in L window
add K' grams in current C window
compare K-grams and K' grams with              subsequent algorithm
if(the identified pattern is already exist in habit file)
increase count of SC-  pattern by 1
else
  {
   Check the pattern in attacker profile
if(Present in profile)
insert SC-pattern into habit file with counter    = 1
else
consider as attack.
   }
  }
 }
```

10. CONCLUSION:

In this paper for the identify SC pattern for the user we have use data mining and forensic technique. Most commonly used SC-patterns are filtered out when the time that a habitual SC pattern appears in the user's log file is counted, and then a user's profile is developed. By detecting a user's SC-patterns as his/her computer usage habits from the user's current input SCs, the IIDPS oppose suspected attackers. The experimental results demonstrate that the average detection accuracy is greater than 94% when the decisive rate threshold is 0.9, indicating that the IIDPS can assist system administrators to point out an internal or an attacker in a closed environment.

REFERENCES:

[1]Techniques Fang-YieLeu, 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" 1932-8184 © 2015
[2] S. Gajek, A. Sadeghi, C. Stuble, and M. Winandy, "Compartmented security for bro wsers Or how to thwart a phisher with trusted computing,"inProc. IEEE Int. Conf. Avail., Rel. Security , Vienna, Austria, Apr. 2007, pp. 120–127.
[3] C. Yue and H. Wang, "BogusBiter: A transparent protection againstphishingattacks,"ACM Trans. Int. Technol., vol. 10, no. 2, pp. 1–31,May 2010.
[4] Q. Chen, S. Abdelwahed, and A. Erradi, "A model-based approach toself-protection in computing system," inProc. ACM Cloud AutonomicComput. 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 adynamic grid-based intrusion detection environment,"J. Parallel Distrib.Comput., vol. 68, no. 4, pp. 427–442, Apr. 2008.
[6] H. Lu, B. Zhao, X. Wang, and J. Su, "DiffSig: Resource differentiationbased malware behavioral concise signature generation,"Inf. Commun.Technol., vol. 7804, pp. 271–284, 2013.
[7] Z. Shan, X. Wang, T. Chiueh, and X. Meng, "Safe side effects commitment for OS-level virtualization," inProc. ACM Int. Conf. AutonomicComput., Karlsruhe, Germany, 2011, pp. 111–120
[8] 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. detection systems: A review,"App[9] J. Choi, C. Choi, B. Ko, D. Choi, and P. Kim, "Detecting web based DDoSattack using MapReduce operations in cloud computing environment,"J. Internet Serv. Inf. Security, vol. 3, no. 3/4, pp. 28–37, Nov. 2013.
[10] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "MIS: Malicious nodesidentificationscheme in network-coding-based peer-to-peer streaming,"inProc. IEEE INFOCOM
, San Diego, CA, USA, 2010, pp. 1–5.

[11] Z. A. Baig, "Pattern recognition for detecting distributed node exhaustionattacks in wireless sensor networks,"Comput. Commun., vol. 34, no. 3,pp. 468–484, Mar. 2011.
[12] H. S. Kang and S. R. Kim, "A new logging-based IP tracebackapproachusing data mining techniques,"J. Internet Serv. Inf. Security,vol.3,no. 3/4, pp. 72–80, Nov. 2013.
[13] K. A. Garcia, R. Monroy, L. A. Trejo, and C. Mex-Perera, "Analyzinglog files for postmortem intrusion detection,"IEEE Trans. Syst., Man,Cybern., Part C: Appl. Rev., vol. 42, no. 6, pp. 1690–1704, Nov. 2012.
[14] M. A. Qadeer, M. Zahid, A. Iqbal, and M. R. Siddiqui, "Network trafficanalysis and intrusion detection using packet sniffer," inProc. Int. Conf.Commun. Softw. Netw., Singapore, 2010, pp. 313–317.
[15] S. O'Shaughnessy and G. Gray, "Development and evaluation of a dataset generator tool for generating synthetic log files containing computerattack signatures,"Int. J. Ambient Comput. Intell., vol. 3, no. 2, pp. 64–76,Apr. 2011.
[16] S. X. Wu and W. Banzhaf, "The use of computational intelligence inintrusion l. Soft Comput., vol. 10,no. 1, pp. 1–35, Jan. 2010.