Post-Attack Intrusion Detection using Log Files Analysis

Apurva S. Patil
Student, Dept. of IT
AVCOE, Sangamner
Maharashtra, India

Deepak R. Patil
Assistant Professor, Dept. of IT
AVCOE, Sangamner
Maharashtra, India

ABSTRACT

Information security is always a main concern of an organization. It is always a challenging job to design a precise Intrusion detection system(IDS) which will detect the intrusions. Intrusion detection systems are broadly classified as host based (HIDS) and network based intrusion detection systems (NIDS). In this paper a comparative study is done on different approaches for detecting intrusion on single host. Point to note that attack detection systems has aim to only detect the activity of intruder and it does not provide any preventive majors.

Keywords

Intrusion, Intrusion detection System, Host based Intrusion detection. Network based Intrusion detection.

1. INTRODUCTION

Intrusion detection systems are one of possible ways to secure the data or information. There are different other ways like cryptography or stenography by which user can share information without compromising. But these approaches are preventive and involve hiding data or encrypting it. An intruder is a legitimate user who makes use of system vulnerability to penetrate the areas of system which are restricted to him and thereby accessing the information. Therefore there is need of something to identify such activities ,Intrusion detection system thus provides us tools to detect such activity. Once the activity detected it is the work of prevention system that the same vulnerability must not used for attack of intrusion again.

1.1 Network based intrusion detection system

Intrusion detection systems are broadly classified as Host based intrusion detection systems and network based intrusion detection. NIDS perform monitoring of network in real time thereby identifying malicious packets and detecting attacks like denial of service, buffer overflow, protocol analysis, CGI attacks.

1.2 Host based intrusion detection system

As the name suggest Host based IDS takes into account the data of single system such as memory buffers , system logs , file system ,various events. It mainly depends on the audit trail data and system call logs for detecting the abnormal behavior . HIDS performs many tasks such as checking memory overflows, detecting malicious behavior of system process and so, There are two approaches for detecting intrusion as Misuse detection and anomaly based intrusion detection Misuse detection.

1.3 Misuse detection

Also called as signature based intrusion detection. It uses predefined rules to detect attacks. The rules look for pattern on network or system operations to find out malicious behavior of network or processes. The only drawback of this approach is the rule file is needed to update regularly for better performance

1.4 Anomaly detection

It is the main area of interest, different approaches that are presented to detect the intrusions are studied. So In anomaly detection the profile for normal behavior of system is created. Then if any malicious activity found it is compared with the normal behavior for detecting attack. There are many factors which are taken in account while profiling normal behavior such as CPU Memory usage, I/O data, login attempts and so on.

Many researchers has contributed in anomaly detection which is elaborated in next section

Starting with the system mentioned in [4], It is a basic intrusion detection system that takes into account system audit trail records, system usage. In [5] a new approach is designed and it was called as IDES . IDES was a real time intrusion detection system which uses a fact that system violations are need to be identified. For that system audit were monitored and abnormal pattern were discovered.

Above two were the detection mechanisms based on system usage. Few more approaches are based on specification of systems. Such as [6] where security critical programs were monitored for vulnerability specification captures the behavior of objects. sequence of operations performed during execution of any program were observed and if it is beyond the specification it is considered as violation. Authors in [7] studied the Linux system calls and developed a system in which system calls are strictly authorized and if the call is not made according to their system call rules it will get called as threat. In [8] authors came with a new system called janus, its main purpose was to secure system that uses untrusted helper applications by monitoring and limiting the system calls it executes. Then few learning based approaches were also suggested which were more efficient. research in this area were conducted on two things first is system call sequence and relationships among arguments of system calls. [9] was a system based on system call sequence called as tide In this normal behavior traces were stored in rule database and later on test traces which might contain attack are compared against the normal. It performs the analysis by using sliding window to look for system call which is ahead and following. In [10] for normal behavior unique sequences of system calls of fixed length are identified and stored in database. for each program a different database was designed. By calculating the number of mismatches the anomalous signal was generated. [11] used sliding window approached for designing Network based intrusion detection system.

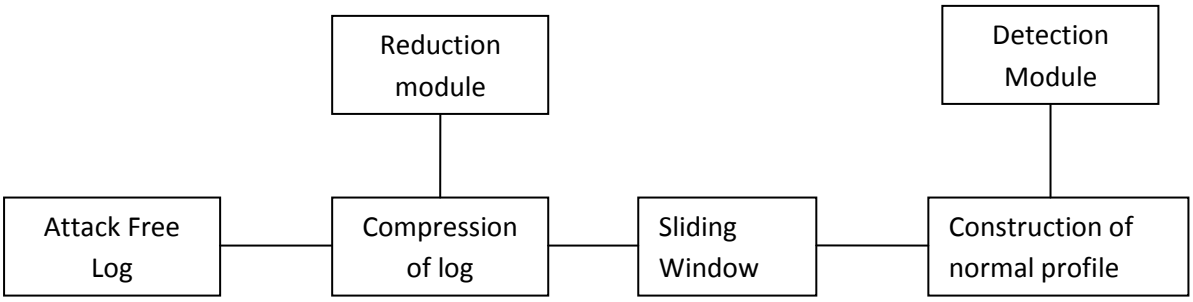

Fig 1. Construction of normal behavior

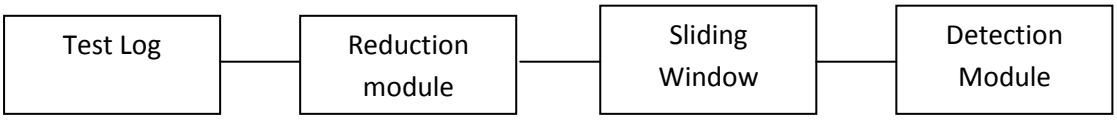

Fig 2. Mechanism of post-attack detection

In [2] HMM were used to detect the abnormal sequences of system calls. Addition to mechanism presented in [10] this system uses frequency of appearance of system calls. For each sequence in the database, tracking of how often it has been seen in the training data is done.[12]also used HMM for the fact that the process behavior has the feature of HMM, and program execution states can be taken as the hidden states in HMM. Because of the performance of HMM authors in [13] introduced a new approach where array of HMM were used for detection of anomalous behavior. authors of [14] proposed detection of intrusion in complete session whereas others were limited to only single application. HMM state were fixed to 6 and window size was fixed to 6. Later on since complete session tracking involved so much volume of data that it was needed that some compression should be applied so author in[15] came with a idea to compress the sequence of system calls and to reduce the log file.

Some researchers used the arguments to system call to find out the malicious activity such as in [16]. Authors proposed that use of complete information is important to avoid mimicry attacks.

Authors in [17] used hierarchical clustering algorithm to model the normal behavior. in this system calls having same arguments were grouped together and the model was built on the cluster representatives instead on direct system calls. Again it was a system designed for only single application. finally in [1] authors came with a system for monitoring entire session based on K-means and HMM.

After studying all the above intrusion detection systems it is found that many different host based anomaly based intrusion detection systems were proposed. Some of them were focusing on single applications and some on sessions.

Furthermore windowing is used in many papers for sliding over the sequences of system call. Compression mechanisms were used in some approaches to reduce the size of system call log data. HMM and clustering algorithms were used in order to build the normal behavior of system. Yet there is scope for improvement by identifying need for better compression algorithm. Finding out what will be the ideal size of window. And results can be compared by improving the clustering algorithm used.

2. PROPOSED MODEL

Here objective is to detect the activity of an intruder after attack and finding out what activity he has done after getting access to system. Here is new model for system which will point to a particular log file which contains attack.

This system follows the learning based approach. Assume that all the activities related to system are monitored and logged in several log files, and behavior of an intruder is significantly different from that of legitimate user. Aim of this system to find out particular activity performed by attacker. This System has to deal with various key issues like huge size of log file and creating a profile of normal behavior only with some part of log file instead of analyzing complete log file.

Two parts in this approach are construction of normal behavior and Mechanism of Post attack detection which are shown in fig 1 and fig 2. To build a profile of normal behavior first selected some attack free log files. Due to large size of system call data a reduction model is used to compress the content of log files. By using a Sequitur to compress the log file. Sequitur [3] generates the rules using grammar. And second part is to detect the actual attack location for that the model uses fuzzy c mean clustering and HMM.

3. CONCLUSION

In this paper different approaches for intrusion detection are presented. Focusing on host based anomaly detection. There were so many different approaches suggested by researchers, few of which were based on single application and some on arguments of system call. Overcoming their drawbacks, new system model is introduced based on HMM and fuzzy c mean algorithm which will hopefully detect location of attack more precisely. It is challenging to get 100% accuracy even if not a single attack free log file is given for training the system.

4. REFERENCES

[1] Karen A. Garc´ıa, Ra´ul Monroy, Luis A. Trejo, Carlos Mex-Perera, and Eduardo Aguirre, "Analyzing Log Files for Postmortem Intrusion Detection,"*in IEEE transactions on systems, man, and cybernetics*, vol. *42, no. 6, november 2012.*

[2] C. Warrender, S. Forrest, and B. A. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Proc. IEEE Symp. SecurityPrivacy*, 1999, pp. 133–145.

[3] C. G. Nevill-Manning and I. H.Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," *J. Artif. Intell. Res.*, vol. 7, pp. 67–82, 1997.

[4] J.-P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Company, Fort Washington, PA, Tech. Rep. 79F296400, Apr. 1980.

[5] D.-E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 222–232, Feb. 1987.

[6] C. Ko, M. Ruschitzka, and K.-N. Levitt, "Execution monitoring of security-critical programs in distributed systems: A specification-based approach," in *Proc. IEEE Symp. Security Privacy*, May 1997, pp. 175– 187.

[7] M. Bernaschi, E. Gabrielli, and V.-L. Mancini, "REMUS: A securityenhanced operating system," *ACM Trans. Inf. Syst. Security*, vol. 5, no. 1, pp. 36–61, 2002.

[8] I. Goldberg, D. Wagner, R. Thomas, and E.-A. Brewer, "A secure environment for untrusted helper applications: Confining the wily hacker," in *Proc. 6th Conf. USENIX Security Symp., Focusing Appl. Cryptogr.*, 1996, vol. 6, pp. 1–13.

[9] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proc. IEEE Symp. Security Privacy*, May 1996, pp. 120–128.

[10] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Security*, vol. 6, no. 3, pp. 151–180, 1998.

[11] W. Lee, C. Park, and S. Stolfo. (1999, Apr.). Automated intrusion detection using NFR: Methods and experiences, in *Workshop on Intrusion Detection and Network Monitoring*, USENIX. Santa Clara, CA [Online].Available:http://www.usenix.org

[12] Y. Qiao, X. Xin, Y. Bin, and S. Ge, "Anomaly intrusion detection method based on HMM," *Electron. Lett.*, vol. 38, no. 13, pp. 663–664, Jun. 2002.

[13] J. Hu, X. Yu, D. Qiu, and H.-H. Chen, "A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection," *IEEE Netw.*, vol. 23, no. 1, pp. 42–47, Jan./Feb. 2009.

[14] F. God´ınez, D. Hutter, and R. Monroy, "On the use of word networks to mimicry attack detection," in *Proc. Int. Conf. Emerging Trends Inf. Commun. Security*, 2006, vol. 3995, pp. 423–435.

[15] N. Wang, J. Han, and J. Fang, "Anomaly sequences detection from logs based on compression," *Comput. Res. Repository*, vol. abs/1109.1729, pp. 1–7, 2011. Available: http://arxiv.org/abs/1109.1729.

[16] C. Kr¨ugel, D. Mutz, F. Valeur, and G. Vigna, "On the detection of anomalous system call arguments," in *Proc. 8th Eur. Symp. Res. Comput. Security*, 2003, vol. 2808, pp. 326–343.

[17] F. Maggi, M. Matteucci, and S. Zanero, "Detecting intrusions through system call sequence and argument analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 4, pp. 381–395, Oct./Dec. 2010.