International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2494-9150 Vol-02, Issue 04, July 2016.

Offline Host Based Intrusion Detection based on Analysis of System Calls

[1]Apurva S. Patil, [2]Dipak R. Patil
[1]Student, [2]Assistant Professor, [1,2]AVCOE, Sangamner, Maharahstra, India.
[1]apurvapatil14@gmail.com, [2]patildipak87@gmail.com

Abstract - Security is always a primary concern of any organization. It is necessary to implement an intrusion detection system (IDS) which will be able to detect the malicious activities over a network or single system. After attack it is important to analyze what intruder has done after getting access to system, what are the areas he tried to penetrate? To identify activity of intruder from huge log file is difficult. Here system is designed, which uses fuzzy c mean clustering along with HMM to build model for ideal behavior of user. Considering the fact that intruder activity pattern is different than normal user a model for detection is built. The input log file is very large therefore sequitur is used to reduce the size of file and windowing is used to process the data efficiently. This system falls under anomaly based intrusion detection system which runs offline to point attack sequence.

Keywords —Intrusion detection system, intruder, anomaly based intrusion detection, fuzzy c mean, sequitur, HMM

I. INTRODUCTION

Intrusion is any activity made to compromise systems integrity, confidentiality, and authentication. So far many intrusion detection systems have been introduced but somehow attacker finds a way to get inside system. Basically attacker finds system vulnerability and tries to exploit. Intrusion detection systems are classified as network based intrusion detection or host based intrusion detection based on its function area, it is also broadly classified as rule based and anomaly based intrusion detection systems based on its detection technique

A. Network Based Intrusion detection systems

In network based intrusion detection data is gathered over a network. It captures network traffic by placing network interface card in promiscuous mode. Then it monitors all the packets that go though the system. Packets content and header information is analyzed in order to find attack or malicious activities. Attacks in network are like buffer overflow, denial of service attack. Installing a network based intrusion detection system will help securing network and thereby avoiding attacks

A. Host Based System

Host based Intrusion detection systems takes into consideration information about single host. Host is single system in a network. It monitors all the activities on a particular system. The source of information for this type of IDS is operating system call logs, events log, audit trails, memory usage, CPU usage, I/O and so on. This helps in detecting what activities user has done after he logged on to system and whether these activities are normal or malicious.

B. Rule Based or signature based Intrusion Detection:

In this type of IDS the analyst looks for particular signature of attack. Signatures are some rules which give explicit indication of particular attack. Signatures of known attacks are stored in database and then compared with input to identify attack.

C. Anomaly based Intrusion detection

Anomaly based Intrusion detection builds a profile of normal behavior, deviation form that is marked as attack. The profile is built on some metric such as number of packet for protocol, traffic rate, port numbers in case of Network based IDS or CPU Memory usage, I/O data, login attempts and so on in case of Host Based IDS. Normal profile is created with attack free data.

The proposed system falls under anomaly based host Based intrusion detection system. It is offline intrusion detection

system whose aim is to identify and point out behavior of an intruder. [3] Gives survey of many IDS. Many researchers have contributed in anomaly detection which is elaborated in next section.

II. LITERATURE SURVEY

Authors of [4] have introduced IDS based on audit trails, how audit trails are used by security officers and some tools which may help security officer to identify intrusions. In [5] real time system was introduced which also processes audit records in order to identify abnormal behavior of user. The profile is built over some metrics such as event counter, interval timer, and resource measure.

In [6] along with audit trail the system uses security specifications for ideal behavior of a program. [7] Is a system where access control database is maintained which has rules. It analyses UNIX system calls deeply and allow only those processes which satisfy the rules for system call and its arguments. Authors in [8] suggested a method based on static analysis. A model is built on control flow of a program from its source code to predict normal behavior.

We are more concerned about the learning based approaches that were introduced following are the few approaches: some of them are based on sequence of system call and some are based on arguments to system call [2], [9] and [10] are methods based on Hidden Markov Model for building profile of normal behavior. HMM had number of states equal to unique system calls executed by a program. Authors in [12] introduced a system based on HMM to build a Model using word network. Each state in word network is HMM model. Authors [11] proposed a system based on series of HMM.

Recently in [13] author has introduced method for mining huge log file and detecting abnormal sequence using grammar based compression. In[16]Statistical analysis of system call argument is done. In [14], [15] Authors proposed approach based on arguments of system calls. Before that the arguments to system call were ignored. After that in [17] model on clusters is built by which system calls having similar arguments were grouped. Author in [1] introduced K-means and HMM based technique for host based anomaly detection. However the accuracy of the system can be improved which is proposed below.

III. PROPOSED SYSTEM

A. *Mechanism*

Input to our system is log files containing sequences of system calls for particular time period. System is offline and purpose of system is to identify whether a particular sequence of system call is normal or it denotes activity of intruder. Here hypothesis is that intruder behavior is considerably different than normal behavior, first a model is built form attack free log and then it is used to identify attack sequence from test log files. Test session has normal as well as attack sequences.

System Architecture is divided in two phases as train and test phase. Following Fig. 1 denotes the train phase of system. This is the process of creating profile for normal behavior.Train log file is given input to the system, it is either attack free log file or attack log file. This file is passed to second stage of compression were its contents are compressed by applying sequitur algorithm. Output of this stage is given to next where a 100 step 100 size window is formed and contents of compressed file are passed through it. In next step a model is built using fuzzy clustering and HMM, which is called detection model.

Fig 2 denotes the test phase of system. In which input given is now combination of attack as well as normal sequences of system calls, which goes through the same sequences as compression and windowing now the input window is checked with the model created form phase 1. If the input file is above a particular threshold then it is considered as abnormal otherwise it is considered as normal sequence.

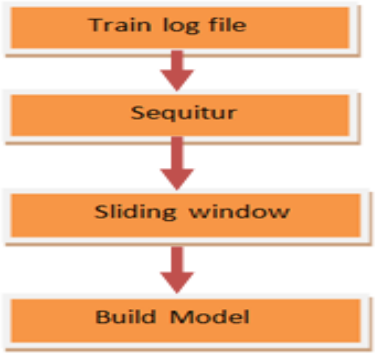

Fig. 1 Construction of profile for normal behaviour

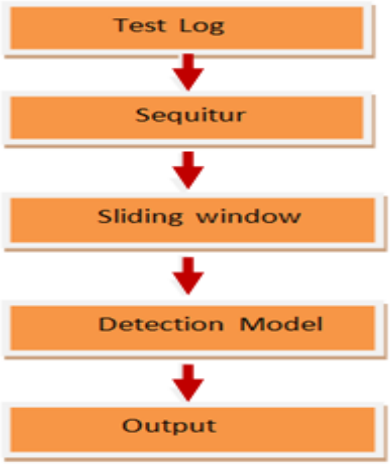

Fig. 2 Detection of abnormal sequence

B. Sequitur

To eliminate the repeating information in log file a grammar based compression technique is used. Following 2 properties ought to be hold so as to cut back the complete size of the grammar.

1. Diagram uniqueness - A pair of adjacent symbols can't appear twice.

2. Rule utilization- A rule generated must be used more than once.

The algorithm scans the input sequence, if it finds repeating symbols it replaces it with a Meta symbol. Following fig. 3 explains sequitur with examples. As one can see in figure 'a' is a simple example of sequitur here S denotes the start symbol and capital letters A,B,C are meta symbols that are generated. In 'c' the input string is such that that there are two possible rules but only one satisfies the first property.

a

Sequence	Grammar
abcdbc	S → aAdA A →bc

b

Sequence	Grammar
abcdbcabcdbc	S →AA A →aBdB B →bc

Sequence	Grammar
abcdbcabcdbc	S →AA A →abcdbc S →CC C →BdA B →aA A →>bc

c

Sequence	Grammar
aabaaab	S →AaA A →aab S →AbAab A →aa

d

Fig 3 Examples

C. Fuzzy C-mean and HMM

1. Fuzzy c mean

It is soft clustering algorithm, in which every data item can be in one or more groups.

After applying sequitur a 100 step 100 size window is slide through the input log file. Then fuzzy c mean is applied and input data is grouped in k-groups. Each cluster has a Member function that denotes degree to which a particular data item belongs to a cluster. One can say due to its fuzzy nature more efficient clustering can be achieved.

2. Hidden Markov Model

Hidden markov model is a doubly stochastic process [18]. It is mostly used in speech recognition applications. HMM assumes that observations are generated by some system whose states are hidden from observer and it also assumes that states satisfy markov property.

More formally HMM has:

1) N- number of states. Note that there are hidden states. i.e. the states that are not directly observable.

2) O- set of observations.

3)A matrix=$\{a_{ij}\}$ that gives the probability of transition from state i to state j.

4)B matrix that gives the probability of emitting a particular observation by a state.

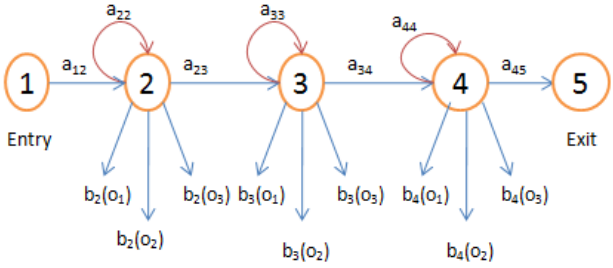

Fig 4. Simple representation of Hmm

In proposed approach after applying sliding window, we have number of windows. First step is to group them in cluster and then apply HMM on group identifiers. Finally calculate average HMM value from all the HMMs and that will be our detection model. Now in test session each window is compared with this detection model to predict whether particular sequence is malicious or not.

IV. RESULTS

For experiment we gathered data from unm dataset for system calls. The log file has sequences of system call along with process ids. For finding out accuracy a test log session is created which had normal as well as attack file for example d1 set had 7 attack files and 3 normal files and likewise we experimentally made few variations,70% attack files 60% attack files and so on. Here's a table that depicts the accuracy measured as detection rate for Fuzzy c mean HMM and k-means HMM.

Method	False Positive (%)	Missing Alarms (%)	Detection Rate (%)
70%			
Fuzzy	2.6	18.0	82.0
KHMM	2.4	19.0	81.0
60%			
Fuzzy	2.3	7.0	81.0
KHMM	2.3	10.0	80.0
50%			
Fuzzy	2.8	19.0	85.0
KHMM	3.5	17.0	83.0
40%			
Fuzzy	2.4	18.0	89.0
KHMM	2.9	21.0	85.0

Table 1: False positive, missing alarms and detection rate comparison

Attack Percentage	Detection rate(%) Fuzzy HMM	Detection rate(%) KHMM
70%	82.0	81.0
60%	81.0	80.0
50%	85.0	83.0
40%	89.0	85.0

Table 2: Comparison of existing and proposed system

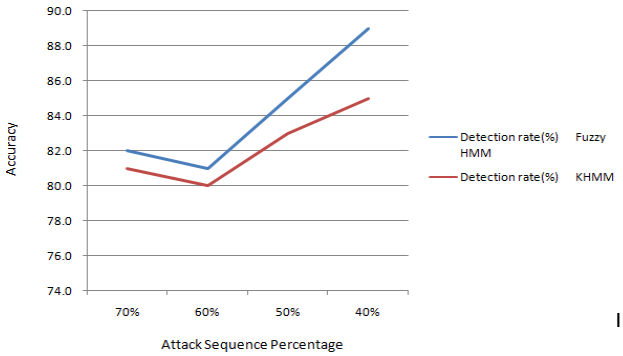

Fig 5. Graph of accuracy

V. CONCLUSION

This offline intrusion detection system helps security analyst to detect the activity of intruder and builds signature of attacks. To identify attack sequences system uses a mode build from Fuzzy clustering and HMM. Fuzzy clustering over data gives better clustering hence the results are improved. Also to reduce the length of huge log file a grammar based approach is used which considerably reduces the log file. Analysing contents by windowing results in accurate pointing to attack sequences. After experimental analysis it can be said that the detection rate i.e. accuracy of system is improved by using fuzzy clustering. However to build profile for normal profile we require at least one attack free log. It will be a big challenge to build a model when there is not a single attack free log available.

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. Security Privacy, 1999, pp. 133–145.

[3] A. Patel, Q. Qassim, andC.Wills, "*Asurvey of intrusion detection and prevention systems*," Inf. Manage. Comput. Security, vol. 18, no. 4, pp. 277– 290, 2010.

[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 security enhanced operating system*," ACM Trans. Inf. Syst. Security, vol. 5, no. 1, pp. 36–61, 2002.

[8] D. Wagner and D. Dean, "*Intrusion detection via static analysis*," in Proc. IEEE Symp. Security Privacy, 2001, pp. 156–168.

[9]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.

[10] D. Yeung and Y. Ding, "*Host-based intrusion detection using dynamic and static behavioral models*," Pattern Recognit., vol. 36, no. 1, pp. 229–243, 2003.

[11]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.

[12] 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.

[13]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

[14] 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.

[15] G. Tandon and P.-K. Chan, "*Learning useful system call attributes foranomaly detection*," in Proc. 18th Int. Florida Artif. Intell. Res. Soc. Conf., 2005, pp. 405–411.

[16] S. Bhatkar, A. Chaturvedi, and R. Sekar, "*Dataflow anomaly detection*," in Proc. IEEE Symp. Security Privacy, 2006, pp. 48–62.

[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.

[18] L.-R. Rabiner, "*A tutorial on hidden Markov models and selected applications in speech recognition*," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286,Feb. 1989.

