International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 09, September-2015

Similarity Measure based Intrusion Detection Systems

Deepapriya U[1]
M.Tech. Computer Networks and Information Security
VNR Vignana Jyothi Institute of Engineering and Technology,
Hyderabad-500090.

Abstract **- This paper introduces methods for detecting anomaly based intrusions using different similarity measures by collecting system calls. These measures examines the frequency and count of common system calls within processes. The KNN classifier which classifies a process as intrusion or not. The training data set which collected during experiments going to compare with the testing DARPA database and evaluated using different similarity measure and its performance is to compare the results of different similarity measures and to find which achieve lower false positive rates at 100% detection rate.**

Index terms: Ids, Anomaly based detection, collection of calls in system, classification algorithm.

1 INTRODUCTION

Extensive use of network computers for essential systems make attraction in computer security. Recent year world is filled with attacks and intrusions. Intrusions are attacks that compromise the principle and undisclosed of the computer by passing the security mechanism. Detection of intrusion is the procedure of scanning the occurrence in the single or in groups of computers and inspect that for anomaly. Increase in attacks in computers makes intrusion detection system to robotize the monitoring and analyzing process allow it to form as one of the essential component in the architecture of computers.

IDS generally consists of three units i.e. Audit Storage, Process and an Alarm/Response unit. The data collected by audit/storage unit analyzed for intrusion signs, processing unit analyzed the data collected from storage unit of audit to detect an attacks if intrusions were found alarm unit sends an alarm or alert to network administrator. Literatures proposed different types of attack prediction systems, it has two types' network and host based. First system collect data from network during transmission and host based collect from the system. Ids further classified into two types depending on their processing unit that is misused and anomaly detection based. The first one compare incoming intrusions with signatures of known intrusions collected in database and later one detect intrusion by monitoring the system behavior if any change occurs it consider as an intrusion. Here we going to deal with novel approach detecting unknown attacks by collecting system calls of each process. In past lot of people had undergone research on this. The researchers at California University conducted experiments on Ids. Davis undergone serious of experiments on anomaly based intrusion detection through with different machine learning algorithms such as cosine metric and RSVM. The former method was simpler and later one shows best results in false positive and detection rate but increase complexity in SVM. The work proposed here was made through inspiration by Sanjay Rawat, Liao and Vemuri paper of intrusion detection systems. The paper describes simple KNN with different similarity measures to find which yields better results.

First in this method we have to observe the execution of process. Any process that contains sequence set of system calls consider as normal process. Alteration in this pattern is called as attack in the structure of abnormality based ids. Here the activity of Ids is to capture the different behavior system calls, which deviates from normal. We use multiple similarity measures in which we measure the analogy among processes that examine two aspects first is existence of calls in system, which are standard among processes and the rate of occurrence of processes system calls. When some execution take place in system it will call the system at some time the two process will call the same system calls present in the trace of these process. More precisely, let x1 ={s1} and x2={s2} are the two distinctive set of system calls term by process x1 and x2 resultantly, where Sj ∈ 'S' is the global system calls. A system call Sj ∈ x1∩x2 then it is common to both process. So we termed this term as Binary weighed cosine metric. Following the methods of Rawat, Liao and Vemuri we also making use of kNN method with new similarity measure to get the better results compared to previous papers.

First of all (a)the Gaussian similarity measure is used to calculate values between the frequency and commonality of system calls(b)then going to classify using kNN classifier for an better detection of intrusions. We are going to prove that by experimental analysis and results.

Section 2 describes about literature survey of previous papers. Section 3 describes about some definitions and knowledge about anomaly detection section 4 describes drawback of previous papers section 5 describes experimental analysis and results section 6 shows comparative study and section 7 shows conclusion and future work of the paper.

2 RELATED WORK

The work of the anomaly ids is to find out anything deviates from the normal process. High false positive rate is one of the common factor of anomaly based ids means it consider even normal process as intrusion and alarm. That's the reason lot of

research activity is going on that. First research was aim to reduce the false positives rates they describe the standard for identifying attacks in computer by supervising system audits. In this perspective description of users were absorbed and comparative methods applied to find out variation from ordinary process. Lane and Broadly suggest different method collecting the users behavior. The database of UNIX command normally issue by users. Any deviation from these behavior specifies intrusion it works good but unable to collect user behavior of larger organization. In another approach normal execution of process is captured this is because for the certain time the execution is normal here the short sequences of system call traced. The same method was taken by Lee et al but they took other approach such as ripper to distinguish normal process ANN are used because it is easy to learn behavior and generalize it lot of bucket algorithms used but it has faults that allows attacker to hide intrusion from Ids. In very reason study neural networks makes use of soundex algorithm which change variable length system call into fixed length and neural network learnt. Due to the drawback of these paper the method was put forwarded by Liao and Vemuri. There they consider each system call as word these all collected to form as documents and every system call change to vectors and applied similarity measure called cosine to find resemblance between the processes based on the KNN classifier.

3 VECTORS AND SIMILARITY MEASURES

Let imagine G be the global system calls from the normal process. From these normal process a matrix C= [cmn] where dmn is the occurrence of mth call in the nth process Where D= [dmn] in binary similarly we have to represent if system call is present then '1' otherwise '0'. Thus for example take two process M1 and M2.

S= {audit, access, close, chdir, create, exit, ioctl, fork}. The detection of system call in two process as follows

M1= {ioctl, close, access, access, exit}

M2 = {audit, ioctl, chdir, access, chdir}

$$A = \begin{pmatrix} 2 & 1 \\ 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

The rows in matrices A and B are the same order in element S and the first matrices contains process X1 as first column and X2 second column contains frequency of system call is presented. The second matrices B represent vectors of '0' and '1' if system call present in set then its represented as '1' else '0'.

3.1 Binary similarity measure

The above formula represent how to calculate binary similarity. Thus the value should be lies between 0 and 1 the μ value increases and decrease depending on the similar system calls present in both the process. Here any new process that is compare with the normal process if the similarity between them equals to 1 then the process is normal or the similarity between them equals to 0 then it classify into abnormal. If the process similarity value which lies more than 0 and less than 1 we have to classify that using KNN algorithm.

3.2 Frequency similarity measure

The similarity score used by Liao and Vemuri was cosine similarity method between two processes is as follows

$$\lambda(M1, M2) = (M1.M2)/|(|M1|)||(|M2|)|$$

$$||M1|| = \sqrt{M1}.\sqrt{M2}$$

3.3 Binary weighed similarity measure

The similarity score used by Sanjay was Binary weighed cosine similarity between two process is as follows

$$Sim(M1, M2) = \mu(M1,M2).\lambda(M1,M2)$$

The motivation of multiplying μ and λ was to get better results compare to previous papers.

3.4 Euclidean Distance

Euclidean distance is one of the most common measure for determining resemblance between two processes is describe below

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

This measures the distance between the processes if the value is low then it is more similar to each other.

3.5 Gaussian Function

Gaussian function is one of the similarity measures used to measure similarity among processes the formula for calculating similarity values is as follows

$$M(Z, Z') = \exp - \left(\frac{\|Z - Z'\|}{2\sigma^2}\right)$$

Where Z1 is mean of Z value and σ is standard deviation of z value.

4 COSINE METRIC WITH K-NN CLASSIFIER

K-NN classifier was used by Liao and Vemuri that is they consider only frequency of system calls instead of ordering. Hence they consider each system call as words and each process as documents using text categorization. They took DARPA data set for experiments a DARPA set is classified into three sets training set, testing test and attack set. Every process

---

If average similarity is greater than threshold of similarity then classify x as standard else intrusion
Specify group of system calls and processes design matrix

X = xij Y = yij take every process P1 from testing data do the following
If system calls from process x not present in global system calls then
X is consider as intrusion
If not
Take every process Aj from training data
Sim calc (P1, Aj)
If calc (P1, Aj) =1.0 then P1 is not intrusion
Exit
Else do
Sort the values and take first biggest k values of calc (P1, Aj)
Compute average similarity of nearest bigger values
If that value greater than predefined threshold value
Then P1 is not attack
Else P1 is intrusion

Of testing set is converted into vectors and compare with each process of training set and find the similarity measures. Here Liao and Vemuri consider only the frequency of system calls so they calculate using cosine similarity measure. If the similarity values are 1 then it will be normal else if its 0 then it will be abnormal. The value lies more than 0 and less than 1 we have calculate kNN by descending highest similarity values and calculate average similarity value by placing highest values(k nearest neighbors) and set threshold if that average similarity value is more than threshold it classify as intrusion or not. But according to Liao and Vemuri paper it shows erroneous results because it calculate only frequency not weight of the system calls. It leads Sanjay Rawat to propose Binary similarity measure in his paper he calculate weight of system calls and multiply cosine similarity and binary weighed similarity to form Binary weighed cosine similarity measure got the better false positive and detection rate compare to previous papers. Later Arun k Poojari extended Binary weighed cosine similarity measure paper by calculating with radial basis function to get better result Sharma also work on this similarity by applying Tanimoto coefficient to get best results.

5 PROPOSED WORK

In this paper we are applying KNN classifier and the different similarity measure in DARPA set. It is classified into three sets training set, testing test and attack set. The DARPA data set was introduced in 1998 to find intrusion here we are labelling training data sets into normal and attack data set by finding the similarity measure between 55 intrusion sessions and training data those training process which exactly match with attack sessions are labelled as abnormal remaining are labelled as normal. Now we form one new labelled training data sets. These labelled are very easy to predict the new incoming process whether as normal or abnormal by calculating similarity measures. Using KNN algorithm we have to set the value of k and sort the distance values in ascending order depending on the value of first k highest values it will classify. BSM audit log files from 1998 DARPA set has been used as testing and training data set in our algorithm. The same data set was used by our previous authors. We extract system call in the following ways. After examine the training data set thoroughly we retrieve 50 unique system calls represented in table 1. For each day a separate BSM file collected contains list of files. The file name ends with zero represent attack file and ends with 1 represent non-attack file. The BSM command such as audit reduce, pr audit and some shell scripts are used to retrieve data that is used in our algorithm. We collect the data for one week the first four day data taken as training data and Remaining days data are collected as testing data. Therefore to compute the effectiveness of our method we took 54 attack data and place that in testing data set

Table 1: List of 55 attacks

1.1 it _ffb_clear, 1.1 _it _format_clear, 2.2 _it_ ipsweep,
2.5 _it_ftpwrite, 2.5 _it_ftpwrite_test, 3.1 _it_ffb _clear,
3.3 _it_ftpwrite, 3.3 _it_ftpwrite_test, 3.4 _it_warez,
3.5 it warezmaster, 4.1 _it_ 080520warezclient,
4.2_it_ 080511warezclient, 4.2_ it_ 153736spy,
4.2_ it_ 153736spy test, 4.2 _it_ 153812spy,
4.4_it_080514warezclient,4.4_it_080514warezclient_test,4.4_it_175320warezclient,
4.4_it_180326warezclient,4.4_it_180955warezclient,
4.4 it 181945warezclient,
4.5 _it _092212_ffb 4.5_it 141011loadmodule,
4.5_it_162228loadmodule,4.5_it_174726loadmodule,
4.5 _it_ format, 5.1_ it _141020ffb, 5.1 _it_ 174729ffb exec,
5.1 _it_ format, 5.2 _it _144308eject clear,
5.2 _it _163909eject clear, 5.3 _it _eject steal, 5.5_ it_ eject,
5.5 _it_ fdformat, 5.5 _it_ fdformat chmod, 6.4 _it_ 090647ffb,
6.4 _it_ 093203eject, 6.4 _it _095046eject, 6.4 _it_ 100014eject,
6.4 _it_ 122156eject, 6.4 _it _144331ffb, test.1.2 format,
test.1.2 format2, test.1.3 eject, test.1.3 httptunnel,
test.1.4 eject, test.1.5 processtable, test.2.1 111516ffb,
test.2.1 format, test.2.2 xsnoop, test.2.3 ps, test.2.3 ps b,
test.2.5 ftpwrite, test.2.4 eject a, test.2.2 format1

Table 2: 50 Global system calls

Setgroups, vfork, unlink, setpgrp, sysinfo, setrlimit, statvfs, stat, su, seteuid, rmdir, setegid, rename, putmsg, readlink, setaudit, pipe, open, oldsetuid, oldnice, oldsetgid, oldutime,pathdonf, login, link, lstat, logout, mkdir, munmap, mmap, mkdir, memcntl, kill,getmsg,ioctl,getaudit, fork, fork1, fcntl, fchown, fchdir, exit,execve, close, creat, chmod, audition, access, audit, chdir, chown.

6 RESULTS

Thus the below table compares the values of different similarity measures if all the similarity measures are showing normal values then obviously that process is normal else attack. So it increases the accuracy of determining intrusion or attack is more

Eucl	Cosine	Binary	BWC	Pearson	Class
9.05	0.944	0.615	0.581	0.912	Normal
308	0.4035	0.310	0..1270	0.2411	Abnormal
57	0.788	0.888	0.700	0.736	Normal
6.4	0.981	0.8	0.872	0.925	Normal

7 CONCLUSIONS

Generally anomaly based intrusion detection system worked on basis of deviation in normal behavior. Here Ids learnt that through analyzing

i) Rate of occurrence
ii) What are the repeated system calls on each process.

In this paper we studied different types of similarity measures and applying on data sets and finding out which one achieving better detection rate and less false positive rate.

ACKNOWLEDGEMENT

We thank Mr. Suresh Reddy, H.O.D of our Department, Mrs. Dr. Mangathaiyaru, coordinator of our Department, Mr. Arun, Assistant professor, for guiding and helping us to present this paper.

REFERENCE

[1] Rawat S, Gulati VP, Pujari AK, Vemuri VR. Intrusion detection using text processing techniques with a binary-weighted cosine metric. Journal of Information Assurance and Security 2006; 1:43–50
[2] S. Axelsson. \Research in intrusion detection systems: A survey". Technical Report No. 98-17, Dept. of Computer Engineering, Chalmers University of Technology, Gteborg, Sweden, 1999.
[3] R. Bace, P. Mell. \NIST special publication on intrusion detection system". SP800-31, NIST, Gaithersburg, MD, 2001.
[4] B. Cha, B. Vaidya, S. Han. \Anomaly Intrusion Detection for System Call Using the Soundex Algorithm and Neural Networks". In Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC'05), pp. 427-433, 2005.
[5] Z. Chan, B. Zhu. \Some Formal Analysis of the Rocchio's Similarity-based Relevance Feedback Algorithm". Technical Report CS-00-22, Dept. of Computer Science, University of Texas-Pan American, Edinburg, TX, 2000.
[6] DARPA 1998 Data, MIT Lincoln Laboratory, http://www.ll.mit.edu/IST/ideval/data/data index.html
[7] D.E. Denning. \An Intrusion-Detection Model". In Proceedings of the 1986 IEEE Symposium on Security and Privacy (SSP '86), IEEE Computer Society Press, pp. 118-133, 1990.
[8] S. Forrest, S. A. Hofmeyr, A. Somayaji, T. A. Longsta. Sense of Self for Unix Processes". In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, IEEE Computer Society Press, pp. 120-128 1996.
[9] S. Forrest, S. A. Hofmeyr, A. Somayaji. Computer Immunology", Communications of the ACM, 40(10), pp. 88-96, 1997.
[10] A. K. Ghosh, A. Schwartzbard. A Study in Using Neural Networks for Anomaly and Misuse Detection".
[11] S. A. Hofmeyr, S. Forrest, A. Somayaji. \Intrusion Detection Using Sequences of System Calls", Journal of Computer Security, 6, pp. 151-180, 1998.
[12] Wenjie Hu , Y. Liao, V. Vemuri. \Robust Support Vector Machines for Anamoly Detection in Computer Security". In International Conference on Machine Learning, Los Angeles, CA. 2003.
[13] T. Lane, C. E. Brodly. \An Application of Machine Learning to Anomaly Detection". In Proceeding of the 20th National Information System Security Conference, Baltimore, MD, pp.366-377, 1997.
[14] W. Lee, S. Stolfo, P. Chan. \Learning Patterns from Unix Process Execution Traces for Intrusion Detection". In Proceedings of the AAAI97 workshop on AI methods in Fraud and risk management, AAAI Press, pp. 50-56, 1997.
[15] Y. Liao, V. R. Vemuri. \Use of K-Nearest Neighbor Classifier for Intrusion Detection", Computers & Security, 21(5), pp.439-448, 2002.