Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 3401 – 3405

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

Hierarchic Clustering Algorithm used for Anomaly Detecting

Zhenguo Chen[*]*, Dongmei Zhu*

Department of Computer, North China Institute of Science and Technology, East Yanjiao, Beijing 101601, China

Abstract

The popularity of using Internet contains some risks of network attacks. Intrusion detection is one major research problem in network security, whose aim is to prevent unauthorized access to system resources and data. This paper choose the clustering algorithm based on the hierarchical structure, to form normal behavior profile on the audit records and adjust the profile timely as the program behavior changed . The algorithm can convert the problem to resolve the problem of massive data processing to the hot research point of anomaly detection. Moreover, in order to improve the results of testing further, we choose data processing algorithm to get high-quality data source. As the experiment shown, we get effective experimental result.

Keywords: Hierarchic Clustering Algorithm; Anomaly Detecting; Intrusion Detection

1. Introduction

It is very important that the security mechanisms of a system are designed so as to prevent unauthorized access to system resources and data. However, completely preventing breaches of security appear, at present, unrealistic. We can, however, try to detect these intrusion attempts so that action may be taken to repair the damage later. This field of research is called Intrusion Detection.

Intrusion detection technology can be divided into two categories: they are misuse detection and anomaly detection .Misuse detection is mainly used of most commercial intrusion detection systems, by matching the current data and signature-known type of attack found. While anomaly detection system compared with the current activities of history normal activities profile, which can detect unknown types

[*] Corresponding author: +8613473641776
E-mail address: czhenguo@gmail.com

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2011.08.637

of attacks. In anomaly detection, we often need to extract the normal historical data of target system. And we use the data generated into the target system normal behavior profile, which use to be as detection system. Anomaly detection still faces many challenges, where one of the most important is the relatively high rate of false alarms (false positives). Many methods [5,7] of clustering are cost higher spatio-temporal consumption.

The model of anomaly detection methods rely on the learning the sample data on training data set. The merits of data source select whether or not are direct impact on the formation of profile about normal system, and the detection result. In this paper, we take the advantages of hierarchical, and it has tree structure. And we choose the training data which comes from the auditing records of solaris system. And we use tf-idf (term frequency-inverse document frequency) weighting algorithm [2] for processing the data of audit records. And we got the test result through the experiment; it proved this method is efficiency in intrusion detection methods.

2. Clustering Algorithm

BIRCH (balanced iterative reducing and clustering using Hierarchics) is a hierarchical incremental clustering method based on the distance, it can real-time adjustment of the cluster profile dynamically. It only occupied small memory through the clustering process, and only scanning once of the data set. This method is suitable for use in large databases as its spatio-temporal is very low.

There are two important concepts in BIRCH algorithm: clustering features (CF) and clustering features tree (CF tree). These two concepts used to be summary for the tree data structure and then gather the information of clusters. The algorithm gets the synthesis description of clusters' information through clustering features (CF), and then to cluster the leaf nodes of the CF-tree. The experiment shows that cluster information received can description CF after the cluster tree contributed sufficiently. The method that extracting data characteristics reducing the memory space which does not reduce the characteristics of data significantly. Given N d-dimensional data points in a cluster:$\{o_i\}$ where i=1, 2,…, N, the Clustering Feature vector of the cluster is defined as a triple: CF=（n，$\overline{LS}$, SS）, where N is the number of data points in the cluster, $\overline{LS}$ is the liner sum of the N data points, i.e., $\sum_{i=1}^{N}\vec{\sigma}$, and SS is the square sum of the N data points, i.e., $\sum_{i=1}^{n}\vec{\sigma^2}$. A CF tree is a height-balanced tree which belongs to the BIRCH algorithm. This tree stores the feature of hierarchical clustering algorithm. Given n d-dimensional data vectors v_i in a cluster $CF_j = \{v_i | i=1\ldots n\}$ the centroid v_0 and radius R (CF_j) are defined as:

$$v_0 = \sum_{i=1}^{N} v_i \tag{1}$$

$$R(CF_j) = \sqrt{\frac{\sum_{i=1}^{n}(v_i - v_0)^2}{n}} \tag{2}$$

R is the average distance from member points in the cluster to the centroid and is a measure of the tightness of the cluster around the centroid.

A fundamental idea of BIRCH is to store only condensed information, denoted cluster feature, instead of all data points of a cluster. Given the CF for one cluster, centroid v_0 and radius R may be computed. The distance between a data point(vector) v_i and a cluster CF_j is the Euclidian distance between v_i and the

centroid, denoted $D(v_i, CF_j)$ while the distance between two clusters CF_i and CF_j is the Euclidian distance between their centroids, denoted $D(CF_i, CF_j)$. If two clusters $CF_i = (n_i, S_i, SS_i)$ and $CF_j = (n_j, S_j, SS_j)$ are merged, the CF of the resulting cluster may be computed as $(n_i + n_j, S_i + S_j, SS_i + SS_j)$.

BIRCH algorithm includes two phases:

The first phase: Scan the data set and establish a initial CF tree which stays in the memory based on BIRCH, which can be regarded as that has been compressed but preserve the cluster structure of the content that inherent data had.

The second phase: To cluster the leaf nodes of the CF-tree leaf by a clustering algorithm.

In our experiment, we select the normal data as training data in the training phase, and we get lots of type of data, so the system normal profile contains lots of type the data information. In the phase of contribute the tree, we can adjust the values of parameters B and M, making the profile of cluster more accurate.

3. Extract Dataset

In intrusion detection system, the information for detection comes mainly from the host system logs, audit records, network data packets, the log data of system for the application and information from other intrusion detection systems or alarm systems monitoring system. Network data packets are mainly used in the misuse detection, testing whether there is attack through pattern-matching or rules-matching. For different users, the log of system application information is also different. So there is no uniform profile to describe such information. But data mining is more effective in dealing with audit records than the data on the network.

Security audit of information system records the information which is security-related in the system. For example, there are records of all the system calls sequence which user-initiated process of implementation in the UNIX system. UNIX security audit records contain a great deal of information about the incident, such as user identify, group information, parameters of system call implementation and return, error codes of system call procedures and implementation. the audit events are mapping into the system call sequence directly by this information. The use of safety audits of the main advantages: it's easy to achieve the classification of the system audit by configuring the audit system. We get the detailed parametric information based on the successes or failures of users, type, event or audit system calls.

ProcessID：994

close execve open mmap open mmap mmap munmap mma mmap close open
mmap close open mmap mmap munmap
mmap close close munap open ioctl access chown ioctl access chmod close
close close close close exit

In a complex and diverse network environment, users and behavior of procedures is changing, but the type of procedures is fixed that implement procedure invoked the private procedure under operating system. When the host network data at the time, the private enforcement procedures call system function, we choose the frequency of the process from the auditing records when call the system function and use them as a training data. In our experiment, we extract the time of the process from system calls of solaris audit logs as training data.

In this paper, we get information of system calls firstly, and use tf-idf frequency weighting method to pretreatment them. We would make pretreatment process mapping for the text classification process, the relationship between the two is: system calls just like "the word", a process seen as a "document." In dealing with text data, the document as a vector which is made of words, all the data both documents and words component a matrix A. what the matrix stored is the frequency of each word in the whole document. i.e. $A = (a_{ij})$, where a_{ij} is the weight of word i in document j. Now let f_{ij} be the frequency of

word i in document j, N the number of documents in the collection, it is the total number of process in my paper. M the number of distinct words in the collection, and it is the total number of system calls which are different from one system call. And n_i the total number of times word i occurs in the whole collection. A simple approach, frequency weighting, use the frequency of word in the document: $a_{ij} = f_{ij}$, but in order to significant difference between the value of normal data and anomaly data, a more common weighting approach is called tf-idf frequency weighting method to calculate a_{ij}:

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{i=1}^{M} f_{ij}^{2}}} \times \log\left(\frac{N}{n_i}\right) \qquad (3)$$

4. Experimental Results

When the normality profile is trained, it is used to detect the unknown data. When a new vector v comes, we calculate the distance from it to the nearest cluster, which means we should search the nearest cluster feature of v from root firstly. We call the radius of the cluster feature R, and we just calculate D(v, v_0)(v_0 is the centroid), v is anomaly if and only if D(v, v_0)>R.

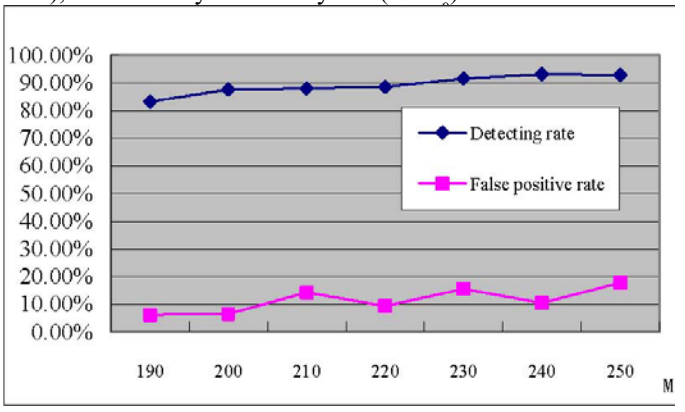

Fig.1 : the effect of M when B=15

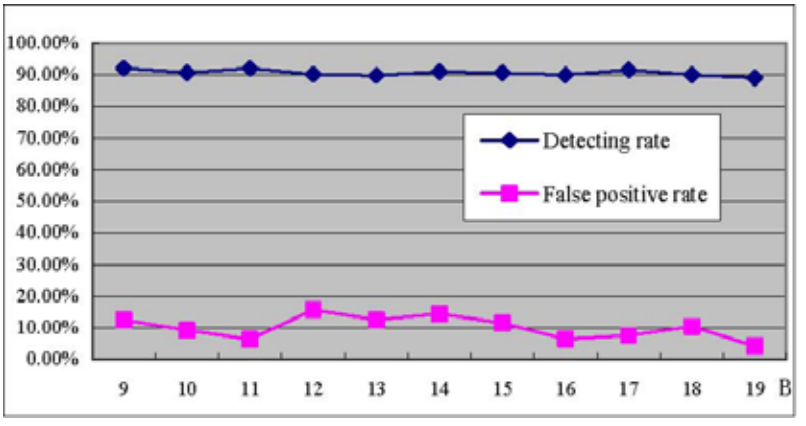

Fig.2 : the effect of B when M=220

In the experiment we select large amounts of data; which training data are information of normal operation of the system call. The data including many kinds of anomalies detected and attack data, making the experiment result more general.

The data we extract if from the DARPA data sets [6], including: a total of 5,046 normal data, from Week 1 of Monday and Tuesday, during the 8:00-16:00. A total of 3,198 abnormal data, from the Week2 Tuesday, Week3 Monday, Week3 Wednesday, Week3 Friday, Week4 Monday. Abnormal data total of 1871 data, from Week6 Thursday.

When testing our training process in DARPA, we test the effect of branch B and cluster number M separately in the hierarchical algorithm. Figure 1 shows the effect of M to the detection rate and false positives rate. Figure 2 shows the effect of B to the detection rate and false positive rate.

We choose M=220, B=11 as the parameters for the detecting of DARPA data set through times of training. We get the average rate of 6.33% false positive, the detection rate was 92.03%. In this paper, we get a higher detection rate and a low rate of false positives of DARPA data sets. So our experiment is feasible and effective that using in intrusion detection based on improved data sources.

5. Summary

In our experiment, we have a lower spatio-temporal cost, the number of participation in clustering of data was 3071, and clustering time for the 1687 ms. Used to detect the normal data was 1975, the detection time was just 407 ms. Data for attack was 3186, and the detection time was 516 ms. The whole process of memory occupied only 1528 k. We also take experiments on KDD data sets, and we make B=15.M=300, we got the detection rate is 99.38%, while the false positive rate is 3.45%. When we change the parameters, the detection rate decreased, but the false positive rate is decreased too. In ADWICE, when the false positive rate is 2.8%, its highest detection positive is 95%.

References

[1] Anderson J.P. Computer security threat monitoring and surveillance. Washington, PA, USA, Technical Report, April 1980.

[2] J.T.Y. Kwok, “Automatic Text Categorization Using Support Vector Machine”, Proceedings of International Conference on Neural Information Processing, 1998.pp. 347-351.

[3] Kalle Burbeck, Simin Nadjm-Tehrani. Adaptive real-time anomaly detection with incremental clustering information. Security technical report 12, 2007. , pp. 56 – 67.

[4] L. Portnoy, E. Eskin, S.J. Stolfo, Intrusion detection with unlabeled data using clustering, in: Proceedings of the ACM Workshop on Data Mining Applied to Security, Philadelphia, PA, 2001.pp. 5-8.

[5] MacQueen, J., “Some Methods for Classification and Analysis of Multivariate Observations,” Proc. 5th Berkeley Symp.,. 1967. pp. 281-297

[6] R. Lippmann, J.W. Haines, D.J. Fried, J. Korba, K.Das. The 1999 DARPA off-line intrusion detection evaluation, Computer Networks 34. 2000. pp.579–595

[7] Sang Hyun Oh, Won Suk Lee. An anomaly intrusion detection method by clustering normal user behavior. Computer and Security Vol22, No.7 2003, pp.596-612.

[8] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "Birch: An Efficient data clustering method for very large databases," Proceedings for the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996. pp. 103-114.