Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp316-320)

Hybrid Model for Computer Intrusion Detection

Witcha Chimphlee[1], Mohd Noor Md Sap[2], Abdul Hanan Abdullah[3], Siriporn Chimphlee[4], Surat Srinoy[5]
[1,4,5]Faculty of Science and Technology
Suan Dusit Rajabhat University
295 Rajasrima Road, Dusit, Bangkok,
Tel: (662) – 244-5225, Fax: (662) – 668-7136 THAILAND
http://www.dusit.ac.th
[2,3]Faculty of Computer Science and Information Systems
University Technology of Malaysia, http://www.utm.my
81310 Skudai, Johor, Malaysia,
Tel: (607) - 5532070, Fax: (607) 5565044 MALAYSIA

Abstract: - The goal of intrusion detection is to discover unauthorized use of computer systems. New intrusion types, of which detection systems are unaware, are the most difficult to detect. In this paper we propose an intrusion detection method that combines *rule induction analysis* for misuse detection and Fuzzy *c*-means for anomaly detection. Rule induction is used to generate patterns from data and finding a set of rules that satisfy some predefined criteria. Fuzzy *c*-Means allow objects to belong to several clusters simultaneously, with different degrees of membership. Our method is an accurate model for handle complex attack patterns in large networks. We used data set from 1999 KDD intrusion detection contest.

Key-Words: - Intrusion Detection System, Induction Analysis, Algorithms, Fuzzy *c*-means, KDD99

1 Introduction

As defined in [1], intrusion detection is "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. It is also defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network". Intrusion detection is an important part of computer system defence. The goal for handle intrusion detection problem is to classify patterns of the system behavior in two categories (normal and abnormal), using patterns of known attacks, which belong to the abnormal class, and patterns of the normal behavior. Without any prior knowledge of attacks classification, if we attempt to divide this set of data into similar groupings, it would not be clear how many groups should be created [2].

There are two main intrusion detection systems. Anomaly intrusion detection system is based on the profiles of normal behaviors of users or applications and checks whether the system is being used in a different manner [3]. The second one is called misuse intrusion detection system which collects attack signatures, compares a behavior with these attack signatures, and signals intrusion when there is a match. Generally, there are four categories of attacks [4]. They are: 1) DoS (denial-of-service), for example ping-of-death, teardrop, smurf, SYN flood, and the like. 2) R2L : unauthorized access from a remote machine, for example guessing password, 3). U2R : unauthorized access to local super user (root) privileges, for example, various "buffer overflow" attacks, and 4) PROBING: surveillance and other probing, for example, port-scan, ping-sweep, etc.

Some of the attacks (such as DoS, and PROBING) may use hundreds of network packets or connections, while on the other hand attacks like U2R and R2L typically use only one or a few connections[5]. IDS can be classified based on the functional characteristics of detection methods as knowledge based intrusion detection and behavior based intrusion detection [6]. The task of an intrusion detection system is to protect a computer system by detecting and diagnosing attempted breaches of the integrity of the system. Anomaly detection still faces many challenges, where one of the most important is the relatively high rate of false alarms (false positives). The problem of capturing a complex normality makes the high rate of false positives intrinsic to anomaly detection except for simple problems [7].

The paper is structured as follows. In section 2 presents rule induction analysis. Section 3 describes a brief fuzzy *c*-means. Explains about experimental design in section 4. Section 5 evaluates our intrusion detection model through experiments. Finally, section 6 presents our conclusion and some discussion.

2 Rule Induction Analysis

Rule or decision tree induction is the most established and effective data mining technologies in use today. The objective is to determine the best set of rules for prediction and classification. It is what can be termed "goal driven" data mining in that a business goal is defined and rule induction is used to generate patterns that relate to that business goal. The business goal can be the occurrence of an event such as "response to mail shots" or "mortgage arrears" or the magnitude of an event such as "energy use" or "efficiency". Rule induction will generate patterns relating the business goal to other data fields (attributes) [15]. The rule induction with C5.0 using Cubit [16] was implemented to provide the rule sets for various categories of attacks and normal data. In this paper, we implemented by improve rule induction using 1R from E. Hooper [17] and using fuzzy *c*-means for handle unknown attacks class. Example of class rules as follows.

2.1 Normal Class Rules

*If Protocol_type = "tcp" And Service = "http" And Flag = "SF" And Land = 0 And Wrong_fragment = 0 And Urgent = 0 And Hot = 0 And Num_failed_logins = 0 And Logged_in = 1 And Num_compromised = 0 And Root_shell = 0 And Su_attempted = 0 And Num_root = 0 And Num_file_creations = 0 And Num_shells = 0 And Num_access_files = 0 And Num_outbound_cmds = 0 And Is_hot_login = 0 And Is_guest_login = 0 And Serror_rate = 0 And Srv_serror_rate = 0 And Rerror_rate = 0 And Srv_rerror_rate = 0 And Same_srv_rate = 1 And Diff_srv_rate = 0 And Srv_diff_host_rate=0AndDst_host_same_srv_rate =1And Dst_host_diff_srv_rate = 0 And Dst_host_srv_diff_host_rate=0 And Dst_host_serror_rate = 0 And Dst_host_srv_serror_rate = 0 And Dst_host_rerror_rate = 0 And Dst_host_srv_rerror_rate =0 Then Class_type = "**normal.**"*

2.2 DoS-Denial of Service Rules

If Duration = 0 And Protocol_type = "tcp" And Service = "http" And Wrong_fragment = 0 And Urgent = 0 And Num_failed_logins = 0 And Logged_in = 1 And Root_shell = 0 And Su_attempted = 0 And Num_root = 0 And Num_file_creations = 0 And Num_shells = 0 And Num_access_files = 0 And Num_outbound_cmds = 0 And Is_hot_login = 0 And Is_guest_login = 0 And Serror_rate = 0 And Srv_serror_rate = 0 And Rerror_rate = 0 And Same_srv_rate = 1 And Dst_host_same_srv_rate = 0 And Dst_host_diff_srv_rate = 0 And Dst_host_srv_diff_host_rate = 0 And Dst_host_serror_rate = 0 And Dst_host_srv_serror_rate = 0 Then ***Class_type = "back."***

2.3 R2L – Remote to Local Rules

If Protocol_type = "tcp" And Land = 0 And Wrong_fragment = 0 And Urgent = 0 And Hot = 0 And Num_failed_logins = 1 And Logged_in = 0 And Num_compromised = 0 And Root_shell = 0 And Su_attempted = 0 And Num_root = 0 And Num_file_creations = 0 And Num_shells = 0 And 12 Num_access_files = 0 And Num_outbound_cmds = 0 And Is_hot_login = 0 And Same_srv_rate = 1 And Diff_srv_rate = 0 And Srv_diff_host_rate = 0 And Dst_host_srv_diff_host_rate = 0 Then ***Class_type = "guess_passwd."***

2.4 U2R – Unauthorized access to Root Rules

If Protocol_type = "tcp" And Flag = "SF" And Land = 0 And Wrong_fragment = 0 And Urgent = 0 And Su_attempted = 0 And Num_root = 0 And Num_outbound_cmds = 0 And Is_hot_login = 0 And Is_guest_login = 0 And Serror_rate = 0 And Srv_serror_rate = 0 And Rerror_rate = 0 And Srv_rerror_rate = 0 And Same_srv_rate = 1 And Diff_srv_rate = 0 And Srv_diff_host_rate = 0 Then ***Class_type = "buffer_overflow."***

2.5 Probing Rules

If Duration = 0 And Land = 0 And Wrong_fragment = 0 And Urgent = 0 And Hot = 0 And Num_failed_logins = 0 And Num_compromised = 0 And Root_shell = 0 And Su_attempted = 0 And Num_root = 0 And Num_file_creations = 0 And Num_shells = 0 And Num_access_files = 0 And Num_outbound_cmds = 0 And Is_hot_login = 0 And Is_guest_login = 0 And Count = 1 And Serror_rate = 0 And Srv_serror_rate = 0 And Rerror_rate = 0 And Srv_rerror_rate = 0 And Same_srv_rate = 1 And Diff_srv_rate = 0 And Dst_host_serror_rate = 0 And Dst_host_srv_serror_rate = 0 And Dst_host_rerror_rate = 0 And Dst_host_srv_rerror_rate = 0 Then ***Class_type = "ipsweep."***

3 Fuzzy c-means (FCM)

Fuzzy c-means (FCM) algorithm, also known as fuzzy ISODATA, was introduced by Bezdek [8] as extension to Dunn's [11] algorithm to generate fuzzy sets for every observed feature. The Fuzzy c-means clustering algorithm is based on the minimization of an objective function called c-means functional.

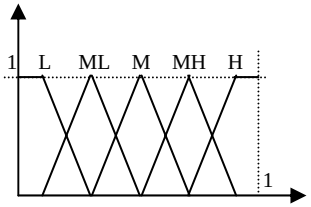

Fig 1: A fuzzy space of five membership function

The fuzzy membership functions corresponding to the informative regions are stored as cases. A collection of fuzzy sets, called fuzzy space, defines the fuzzy linguistic values or fuzzy classes. A sample fuzzy space of five membership function is shown in Fig 1.

Fuzzy clustering methods allow for uncertainty in the cluster assignments. FCM is an iterative algorithm to find cluster centers (centroids) that minimize a dissimilarity function. Rather that partitioning the data into a collection of distinct sets by fuzzy partitioning, the membership matrix (U) is randomly initialized according to Equation 1.

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall_j = 1, ..., n. \tag{1}$$

The dissimilarity function which is used in FCM in given Equation

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}{}^m d_{ij}{}^2 \tag{2}$$

uij is between 0 and 1;
ci is the centroid of cluster i;
dij is the Euclidian distance between ith centroid (ci) and jth data point;
$m \in [1, \infty]$ is a weighting exponent.
To reach a minimum of dissimilarity function there are two conditions. These are given in (3) and (4).

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}{}^m x_j}{\sum_{j=1}^{n} u_{ij}{}^m} \tag{3}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \tag{4}$$

Detailed algorithm of fuzzy c-means proposed by Bezdek in 1973 [8]. This algorithm determines the following steps in Fig 2.

Algorithm . Fuzzy *c*-means
Step 1: Randomly initialize the membership matrix (U) that has constraints in Equation 1.
Step 2: Calculate centroids (c_i) by using Equation 3.
Step 3: Compute dissimilarity between centroids and data points using Equation 2. Stop if its improvement over previous iteration is below a threshold.
Step 4: Compute a new U using Equation 4 go to step 2.

Fig 2: Fuzzy c-Means Clustering [8].

By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the "right" location within a data set.

FCM does not ensure that it converges to an optimal solution. Because of cluster centers (centroids) are initializing using U that randomly initialized. (Equation 3).

Performance depends on initial centroids. For a robust approach there are two ways which is described below [13].
1.) Using an algorithm to determine all of the centroids. (for example: arithmetic means of all data points)
2.) Run FCM several times each starting with different initial centroids.

4 Experimental Design

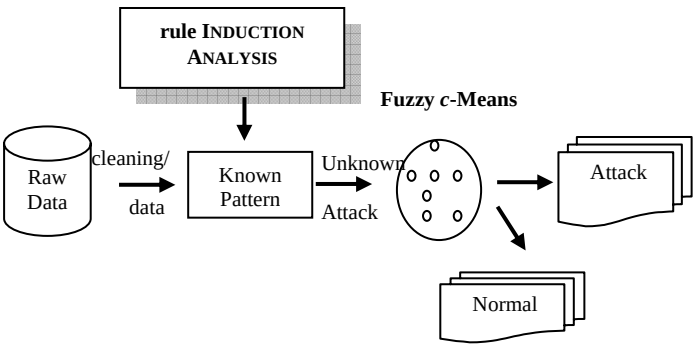

Fig 3: Step for detection

In our method have three steps (Fig.3). First step for cleaning (handle missing and incomplete data). Second step for analysis has known pattern using induction analysis and the last step for detecting attack pattern using fuzzy c-means. The first step involved mapping symbolic-valued attributes to numeric-valued attributes and the second step implemented non-zero numerical features.

5 Experimental setup and results

In this experiment, we use a standard dataset the raw data used by the KDD Cup 1999 intrusion detection contest [12]. This database includes a wide variety of intrusions simulated in a military network environment that is a common benchmark for evaluation of intrusion detection techniques. In general, the distribution of attacks is dominated by probes and denial-of-service attacks; the most interesting and dangerous attacks, such as compromises, are grossly under-represented [13]. The data set has 41 attributes for each connection record plus one class label. There are 24 attack types, but we treat all of them as an attack group. A data set of size N is processed. The nominal attributes are converted into linear discrete values (integers). After eliminating labels, the data set is described as a matrix X, which has N rows and m=41 columns (attributes). There are md=8 discrete-value attributes and mc = 33 continuous-value attributes.

We ran our experiments on a system with a 1.5 GHz Pentium IV processor and 512 MB DDR RAM running Windows XP. All the preprocessing was done using MATLAB®. MATLAB's Fuzzy Logic Toolbox [14] was used for Fuzzy c-means clustering. In practice, the number of classes is not always known beforehand. There is no general theoretical solution to finding the optimal number of clusters for any given data set. We choose k = 5 for the study. We will compare five classifiers which have been also used in detecting these four types of attacks.

A considerable amount of data-preprocessing had to be undertaken before we could do any of our modeling experiments. It was necessary to ensure though, that the reduced dataset was as representative of the original set as possible. The test dataset that previously began with more than 300,000 records was reduced to approximately 18,216 records. Table 1 shows the dataset after balanced among category for attack distribution over modified the normal and other attack categories. Preprocessing consisted of two steps. The first step involved mapping symbolic-valued attributes to numeric-valued attributes and the second step implemented non-zero numerical features.

Table 1. Dataset for attack distribution

Attack Category	% Occurrence	Number of records
normal	31.64	5,763
probe	11.88	2,164
DoS	19.38	3,530
U2R	0.38	70
R2L	36.72	6,689
Summary	100	18,216

After we used rule induction analysis for matching pattern of known data can reduce amount of data set that remain only unknown patterns. We finished experiments with small data sets for unknown pattern or anomaly detection. In fuzzy *c*-means stage FCM is able to deal more effectively with outliers and to perform membership grading. In Table 2 display result using fuzzy *c*-means for clustering data from table 1.

Table 2. Results from using fuzzy *c*-means.

Class type	No. of record	Hit	miss	% accuracy
normal	5763	5749	14	99.757
probe	2164	2164	0	100
DoS	3530	2897	634	82.045
U2R	70	67	3	95.714
R2L	6689	6145	544	91.867
Summary	18216	17022	1195	

6 Conclusion

Most intrusion detection systems rely on pattern matching operations to look for attack signatures. Network based intrusion detection are the most deployed IDS. They frequently rely on signature matching detection method and anomaly detection.
We show in this paper that combine misuse and anomaly detection by using rule induction analysis and fuzzy *c*-means. Rule induction is used to generate patterns from data. Fuzzy c-Means allow objects to belong to several clusters simultaneously, with different degrees of membership. We employ data from the third international knowledge discovery and data mining tools competition (KDDcup'99) to train and test feasibility of our proposed model.

From our experimental results our model achieves more than 93 percent detection rate and

less than 7.76 percent false alarm rate for five typical types of attacks. This method was efficient and reduce amount of data set for handle data and we build the model to improve the detection rate. Future work, we plan to extend this system to operate in a high accurate and low false alarm rate with unlabeled data.

References:

[1] R. Bace and P. Mell, Intrusion Detection Systems, *NIST Special Publications on Intrusion Detection System*, 31 November 2001.

[2] D. Denning, An intrusion-detection model, *In IEEE computer society symposium on research in security and privacy*, 1986, pp. 118-131.

[3] H. Jin, J. Sun, H. Chen, and Z. Han, A Fuzzy Data Mining Based Intrusion Detection System, *Proceedings of 10th International Workshop on future Trends in Distributed Computing Systems (FTDCS04) IEEE Computer Society*, Suzhou, China, May 26-28, 2004, pp. 191-197.

[4] W. Lee, S. Stolfo, and K.Mok, A data mining framework for building intrusion detection models, *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, May 1999, pp. 120-132.

[5] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar, A comparative study of anomaly detection schemes in network intrusion detection, *In SIAM International Conference on Data Mining*, 2003.

[6] B. Balajinath, S.V. Raghuvan, *Intrusion detection through learning behavior mode*l, 2001.

[7] K. Burbeck and S. Nadjm-Tehrani, Adaptive Real-Time Anomaly Detection with Improved Index and Ability to Forget, *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops*, June 2005.

[8] J. Bezkek, Pattern Recognition with Fuzzy Objective Function Algorithms", *Plenum Press*, USA, 1981.

[9] S. Albayrak, Fatih Amasyali, Fuzzy c-means clustering on Medical Diagnostic Systems, *International XII Turkish Symposium on Artificial Intelligence and Neural Networks*, 2003.

[10] F. Godínez, D. Hutter, R. Monroy. Attribute Reduction for Effective Intrusion Detection, *AWIC 2004*, 2004, pp. 74-83

[11] J.C. Dunn, "A Fuzzy Relative of the ISODATA process and its Use in Detecting Compact", Well Separated Clusters, Journal of Cybernetics, Vol. 3, No.3, 1974, pp. 32-57.

[12] KDD data set, 1999; http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[13] P. Laskov, K. Rieck, C. Schäfer, K.R. Müller, "Visualization of anomaly detection using prediction sensitivity", Proceeding of Sicherheit, April 2005, pp. 197-208.

[14] Math Works, *Statistical Toolbox for User's Guide*, *Math Works*, 2001.

[15] http://www.intellicrafters.com/algorithms.htm

[16] Rule Induction with C5.0, See5/Cubit software. Rulequest Research, 1997-2002.

[17] E. Hooper, the Derivation of an Efficient and Accurate Methodology for Intrusion Detection. PhD research, Department of information systems, University of East Anglia.