Suppressing False Alarms of Intrusion Detection Using Improved Text Categorization Method

Zonghua Zhang and Hong Shen
Graduate School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Tatsunokuchi, Ishikawa, 923-1292, Japan
E-mail:{zonghua, shen}@jaist.ac.jp

Abstract

Although some text processing techniques can be employed to intrusion detection based on the characterization of the frequencies of the system calls executed by the privileged programs, and achieve satisfactory detection accuracy, high false alarms make it hardly practicable in real life. In this paper, we modified the traditional weighting method tf-idf for suppressing false alarms by considering the necessary information between the processes and sessions. Preliminary experiments with 1998 DRRPA BSM auduit data show that our modified method can suppress high false alarms effectively while maintaining satisfactory detection accuracy, which thus make text categorization approaches more practicable for intrusion detection.

1. Introduction

It is well known that intrusion detection can be formulated as a text categorization problem, and it outperforms the approaches from machine learning, data mining, and pattern recognition in terms of ease of use and less computational overhead. Liao et al [4] ever applied K-Nearest Neighbor(KNN) classifier to label program behavior as normal or intrusive. Specifically, each system call in the process is treated as a word and the collection of system calls over each program execution as a document, and thus the system call frequencies are taken as characteristic to represent program behavior. Although the text processing model could achieve satisfactory detection accuracy, high false alarm rate make it hardly meet demands in real life. With the objective in mind and after analyzing the generation of false alarms, we propose another modified weighting model based on the tf-idf weighting method. In our model, the characterization of the system calls are not only depend on the processes, information and attributes about the sessions are also considered. Training data and testing data are analyzed in the unite of processes, and the relationship between the processes are further analyzed by considering the prerequisite and consequence together with time information related with them, especially when a process is determined as anomaly.

The paper is organized as follows. Section 2 gives the brief introduction of tf-idf weighting method to intrusion detection. Section 3 proposes our modified weighing model and introduces two candidate methods for intrusion detection. Section 4 provides the experiment implementation together with the analysis and discussion of the result, and we conclude in section 5.

2. Existing Weighting Method

When intrusion detection at the level of privileged process is formulated as text categorization problem, each system call of the process is treated as a "word" and the set of system calls generated by a process is regarded as the "document" [4]. The simple frequency weighting method and tf-idf weighting method can be applied to transfer a process generated by the program into a vector. The simple model was established as follows:

Matrix $A = a_{ij}$, the collection of processes from different sessions, and a_{ij} is the weight of system call i in process j.

f_{ij}, the frequency of system call i in process j.

N, the number of processes in the collection.

M, the number of distinct system calls in the collection.

n_i, the number of times that system call i appears in the collection.

Thus, frequency weighting is defined as:

$$a_{ij} = f_{ij} \tag{1}$$

Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)
0-7695-2073-1/04 $20.00 © 2004 IEEE

$tf\text{-}idf$ weighting method is defined as:

$$a_{ij} = \frac{f_{ij}}{\sqrt{\sum_{l=1}^{M} f_{lj}^2}} \times log(\frac{N}{n_i}) \quad (2)$$

Henceforth, statistical methods, classification methods and machine learning techniques can all be applied to distinguish normal behaviors and anomalies based on the transferred vector space model.

3. Modified Weighting Methods for the Characterization of UNIX Processes

In [4], processes were deal with independently, and a session is labelled as intrusive once one of its process is detected as anomaly, which thus increase the possibility for the generation of false alarm. Additionally, information related to the time are ignored, so does the correlation between processes from the same session.

3.1. Session-specific Weighting Method

When a connection is established between two hosts, several sessions or processes will be generated subsequently, in order to reflect the source specific differences, we add the information of the session (such as Source Machine or session ID, which can be regarded as the topic of documents [1]). Accordingly, the $tf\text{-}idf$ model can be improved as following: $pf_{s,t}(\theta)$ represents the process p from session s at time t which includes system call θ, and it is updated according to the equation:

$$pf_{s,t}(\theta) = pf_{s,t-1}(\theta) + pf_{s,P_t}(\theta) \quad (3)$$

here, $pf_{P_t}(\theta)$ denote the process frequencies in the newly added set of processes P_t. The process frequencies can be used to calculate weights for the system calls θ in the process p. The model based on the fact that different sessions include different processes, and various processes have various system calls, consequently , it reflects session-specific differences. The same system call may have different weights because of different sessions it belongs to. To specific the equation(2), the weight of the system call θ in the processes p can be calculated as follows at time t:

$$w_t(\theta, \overrightarrow{p}) = \frac{(1 + log_2 f(\theta, \overrightarrow{p})) \times log_2(N_t/n_\theta)}{Z_{\overrightarrow{p}}} \quad (4)$$

where

$f(\theta, \overrightarrow{p})$, the frequency of system call θ in the process p;
N_t is the number of processes in the current training set;
n_θ is the number of processes include system call θ;
$Z_{\overrightarrow{p}} = \sqrt{\sum_{\theta \in \overrightarrow{p}} w_t(\theta, \overrightarrow{p})^2}$ is the 2-norm of vector $\overrightarrow{p}$.

When calculate the weights of the system calls, we apply the session-specific $pf_{s,\theta}$ instead of pf_θ. Therefore, information about the session should be carried in our method. If no training data is available at $t = 0$ for a specific session, we can set $pf_{s,0} = 0$ for its all θ or identify other similar sessions $s^{'}$, that is, $df_{s,0}(\theta) = \sum_{s'} pf_{s',0}(\theta)$. This case happens when an anomaly detector is trained online.

3.2. Additional Measures

Brants et al [1] ever proposed several methods to improve the $tf\text{-}idf$ model for detecting new events. According to the characteristic of our intrusion detection model, we apply and modify some of these methods.

Firstly, the similarity between two processes can be measured using cosine distance $cos(\overrightarrow{p_1}, \overrightarrow{p_2})$ or Hellinger distance($H(\overrightarrow{p_1}, \overrightarrow{p_2})$) after being transformed into vectors:

$$cos(\overrightarrow{p_1}, \overrightarrow{p_2}) = \sum_{\theta} w_t(\theta, \overrightarrow{p_1}) \cdot w_t(\theta, \overrightarrow{p_2}) \quad (5)$$

$$H(\overrightarrow{p_1}, \overrightarrow{p_2}) = \sum_{\theta} \sqrt{w_t(\theta, \overrightarrow{p_1}) \cdot w_t(\theta, \overrightarrow{p_2})} \quad (6)$$

Secondly, due to the fact that the number of intrusive UNIX processes are much smaller than that of normal ones, the similarity between intrusive processes and the similarity between normal processes should not have the same weight. To capture this difference, the average similarity of the current process $\overrightarrow{p_1}$ to all existing processes from the same session can be calculated:

$$cos^{'}(\overrightarrow{p_1}, \overrightarrow{p_2}) = cos(\overrightarrow{p_1}, \overrightarrow{p_2}) - cos_{avg}(\overrightarrow{p_1}) \quad (7)$$

$cos(\overrightarrow{p_1}, \overrightarrow{p_2})$ can be replaced by $H(\overrightarrow{p_1}, \overrightarrow{p_2})$ if we use Hellinger distance. This step can be omitted if the training data are only normal processes.

Thirdly, based on the fact that the number of system calls in the various processes might different, and inspired by the work [**?**], we divide one process into several segments by a sliding window of fixed length w. For a process with length l, $\lfloor \frac{l-w}{s} + 1 \rfloor$ segments can derive from it, and we assume that minimal occurrence of some attacks can be detected in $[P_i, P_{i+w}]$. The similarity of two processes is the maximum of the similarities between the segments of these two processes, that is:

$$cos^{''}(\overrightarrow{p_1}, \overrightarrow{p_2}) = \max_{\overrightarrow{s_1} \in \overrightarrow{p_1}, \overrightarrow{s_2} \in \overrightarrow{p_2}} cos^{'}(\overrightarrow{s_1}, \overrightarrow{s_2}) \quad (8)$$

where s_1 and s_2 are the segments in processes p_1 and p_2, and $cos^{'}$ is the similarity score from equation (7). In addition, normal processes and abnormal processes in the training data should be updated frequently in real life for restraining false alarms and detecting novel attacks. Therefore, some time information should be considered. Here, we

apply a linear time model [3], and we only consider the similarity within the time window m:

$$cos_{TW}(\overrightarrow{p_1}, \overrightarrow{p_2}) = (1 - time/m) \cdot cos^{''}(\overrightarrow{p_1}, \overrightarrow{p_2}) \quad (9)$$

the similarity to processes outside the window is 0. Of course, at the beginning of the training, time window m should large enough to include all the processes, with the add of the new processes, m can be adjusted manually.

Finally, in order to minimize the false positive rate, when a processes is identified as intrusive, we do not hasty to treat the session it belongs to as an intrusion. As described in [5], the attacks have causal relationship between each other, and the correlation of the attacks could be formulated as a connected DAG(directed acyclic graph) $HG = (N, E)$, where the set N of nodes is a set of attacks, and for each pair of nodes $n_1, n_2 \in N$, there is a edge from n_1 to n_2 in E iff n_1 prepares for n_2. Therefore, the triple $(fact, prerequisite, consequence)$ holds for some stealthy attacks. Based on this assumption, when detecting an intrusive processes, the current session are not taken as intrusion immediately. Its neighbor processes or sessions are also considered carefully.

3.3. Candidate Methods for ID

Many techniques have been applied in text categorization, such as Bayesian Networks, decision trees, neural networks, support vector machines, k-nearest neighbor approach, etc. In this paper, we only apply One-class SVM and KNN as our approaches for testing the new model presented in the last section.

The idea of KNN is very simple, that is, when a new data is available, the similarity between it and each of the training data is calculated using $cosin$ distance or $Hellinger$ distance. The similarity score are sorted and the k nearest neighbors are selected to determined the attribute of the new process. A threshold should be set empirically in order to calculate the average similarity value of the sorted k nearest neighbors.

One-class SVM is one modified SVM, which identifies "outliers" amongst positive examples and use them as negative examples. After mapping between input data space X and high-dimensional feature space H via a kernel, origin is treated as the only member of the second class. Then "relaxation parameters" is used to separate the point of the first class from the origin.

4. Experiment

In order to compare our modified weighting model with the original $tf\text{-}idf$ model, we also selected the 1998 $DARPA$ data [2] as our experiment data, which consists seven weeks of training data and two weeks of testing data.

4.1. Training and Testing Data

Four days were picked arbitrarily out of five days that free of attacks for training, and the left one for the normal part of testing data. There are 606 distinct processes drawn from more than 2000 sessions running on the victim Solaris machine during the selected four training days, and 55 intrusion sessions in the other seven-week training data.

	Data Set I (process)		Data Set II (process)	
	normal	intrusive	normal	intrusive
Training	500	0	500	20
Testing	5285	35 attacks	5285	24 attacks

Table 1. Experiment Data Sets

Two data sets for the experiment were formulated. Data set I consists only normal processes, and data set II consists both normal processes and anomaly processes. 500 out of the 606 distinct processes were selected as the training set of Data set I. The 3rd day of the 7th training week which contains 412 sessions and 5285 normal processes was chosen as normal part of the first testing data set (some of them are same in order to count false alarms), while abnormal part of testing data contains all the 55 intrusive sessions. Additionally, there are 30 distinct processes from 55 intrusive sessions, and 20 of them are incorporated into the first training data set to form the second training data set. 24 attacks within the two-week DARPA testing audit data are incorporated into the 5285 normal processes to form the second testing data set. According to our definition, each attacks counts as one detection, even with multiple sessions. The detection accuracy is then calculated as the rate of detected attacks, and the false positive rate is defined as the probability of misclassified normal processes(these two terms are not rigorous symmetry here). The training data and testing data are illustrated in Table 1.

4.2. Results and Discussion

One-class SVM and KNN were applied to the training data set and ROC was employed to describe the performance. For KNN methods, the number of nearest neighbors of the test processes, k, and the threshold, which is the average similarity of the k nearest neighbors affect its performance, and here it is induced by changing the similarity threshold, while One-class SVM is induced by the various of v and σ(σ is the variable of the RBF kernel function). The similarity is measured by Hellinger distance. The

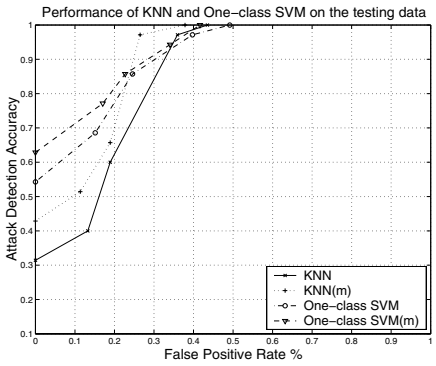

Figure 1. Comparison of the Performance I

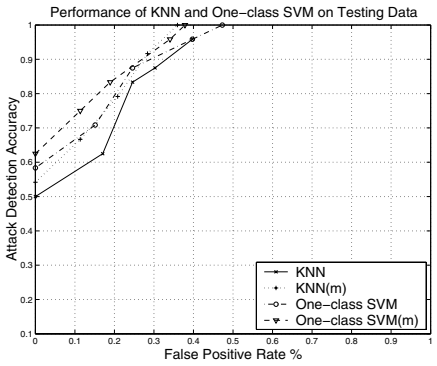

Figure 2. Comparison of the Performance II

result using experiment data set I can be illustrated by figure 1(KNN(m) means KNN method based on our modified weighting model), and as the figure shows, the performance of KNN and One-class SVM has significant improvement using modified weighting method than using original tf-idf weighting method. The value of k varies from 5 to 30, and we select the best performance(when k=10) of this range for showing. Additionally, keeping false alarm rate as 0, KNN/One-class SVM can detect 11/19 out of 35 attack instances using original tf-idf weighting method, while 15/22 out of 35 attack instances using modified weighting method. To acquire the result by One-class SVM, $v = 0.0001$ and $\sigma = 12$ when using original tf-idf model, while $v = 0.0001$ and $\sigma = 14$ when using modified weighting method.

In the second experiment, the new available process is compared with the intrusive processes firstly. Compared with KNN method, One-class SVM had no great change. In fig 2, the value of k is 10. For One-class SVM, we only adjusted the value of v and σ to achieve the satisfied performance. When keeping false alarm rate as 0, the detection rate can achieve 62.5%(15 out of 24 attacks, $v = 0.01$ and $\sigma = 12$), and the detection rate reaches 100% rapidly while false positive remains 0.38%(20 out of 5285 normal processes). It is worth noting that original KNN can not detect the DoS attack *process table* because of the normal appearance of the processes issued by this session, but KNN using our modified model can detect it easily with a lower false alarm rate. In addition, the time window T in our weighting model disclose the attempt of the *process table* attack that intends to exhaust the process table of the victim machine by establishing connection every several seconds.

5. Concluding Remarks

Based on the special characteristics of the observable subjects, we modified the original tf-idf weighting model with the consideration of the necessary information. The experiments show that our weighting model can suppress false alarms significantly compared with the original weighting method. However, it seems that there is still a long way to meet the requirements in real life. But actually, we can not exclude the reason form the limited sample of the testing data. Furthermore, we can conclude that the characterization of the observable subjects is more important than the specific method, so the effective description of the subjects rather than the method is more meaningful for improving the performance of intrusion detection that using text processing techniques.

Acknowledgment

This research is conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology.

References

[1] Thorsten Brants, Francine Chen, Ayman Farahat, "A System for New Event Detection," *SIGIR'03, July 28-August 1, 2003,* Toronto, Canada.

[2] MIT Lincoln Laboratory, *http://www.ll.mit.edu/IST/ideval/.*

[3] Peng Ning, Yun Cui, Douglas S.Reeves, "Constructing Attack Scenarios through Correlation of Intrusion Alters," *CCS'02, Noverber 18-22, 2002,* Washington,DC, USA.

[4] Y.Liao and V.R.Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection," *Computers & Security,* 21(5), pp439-448, Oco,2002.

[5] Y.Yang, T.Pierce, and J.Carebonell. "A sudy on retrospective and on-line event detection," *In proceeding of SIGIR-98,* Melbourne, Australia, 1998.