A novel anomaly detection Based on Vector Quantization

Qingbo Yin, Rubo Zhang, Xueyao Li, Liran Shen

College Of Computer Science & Technology, Harbin Engineering University, Harbin 150001, People's Republic of China

(Email: yinq@0451.com)

Abstract

Intrusion detection has emerged as an important approach to network security. A new method for anomaly intrusion detection is proposed based on the Splitting vector quantization. The splitting vector quantization is employed to map short sequences of system calls of the privileged processes to the disjoint feature subspaces. The relationship among these vector subspaces is used to classify the normal or abnormal. The observed behavior of the system is analyzed to infer the probability. The experiments have showed that the method is effective to detect anomalistic behaviors, and enjoys better generalization ability.

Keywords: Intrusion detection, Splitting Vector Quantization, Markov chain model

1. Introduction

As the increase of the significance of computer network in modern society, its security becomes one of the hottest issues to be solved. Therefore, it is extremely imperative to find an effective way to protect this valuable network infrastructure. In general, intrusion detection may be defined as "the problem of identifying individuals who are using a computer system without authorization (i.e., 'crackers') and those who have legitimate access to the system but are abusing their privileges (i.e., the 'insider threat')" [1]. An ideal intrusion detection system is the one that has 100% attack detection rate along with 0% false positive rate (the rate of mis-classified normal behavior), requires light load of monitoring, and involves minor calculation or overhead. Current intrusion detection systems, however, are plagued by either high false alarm probability or low attack detection accuracy.

Intrusion detection systems (IDS) are generally categorized as misuse detection and anomaly detection. Misuse detection via signature verification compares a user's activities with the known signatures of attackers attempting to penetrate a system. While misuse detection is useful for finding known intrusion types, it cannot detect novel attacks. Unlike misuse detection, anomaly detection identifies activities that deviate from established statistical patterns for users, systems or networks. In spite of their capability of detecting unknown attacks, anomaly detection systems suffer from high false alarm rate when normal user profiles and system or network behavior vary widely.

In this paper, a new method is proposed to learn program behavior in intrusion detection. The Splitting Vector Quantization (SVQ) is used to map the short system calls sequences to the disjoint action subspaces. Afterward, The relationship among these vector subspaces is used to classify the normal or abnormal. The observed behavior of the system is analyzed to infer the probability.

2. Principles and Method

2.1 Principles

A large number of anomaly detection methods have been used to identify anomalous activities, such as neural network, artificial immune systems, machine learning and others [2-17].

Anomaly detection can be combined with signature verification to detect attacks more efficiently. The biggest challenge is to choose features that best characterize the user or system usage patterns so that non-intrusive activities would not be classified as anomalous. Clearly, the inclusion of too much data will adversely impact the performance of the system, while the inclusion of too little data will reduce the overall effectiveness of the system.

Most of anomaly detection approaches build models of normal data, and then detect deviation from the normal model in observed data. These methods need large numbers of purely normal data from which they train their model. In practice, it is impossible that large number of purely normal data readily available were achieved. But, it is sure that a small number of purely train dataset can be obtained. To solve this difficulty, we need a technique that enjoys the better generalization ability when a small number of train dataset are used only.

Since many intrusions are composed of a series of related computer actions, the temporal profile of action sequence is important to detect intrusions. The norm profile of temporal behavior should capture the temporal dependency among computer actions during the normal usage of a computer and network system [4]. In the past, we dealt with short system call sequences as features. But the relationship between the features wasn't used. So the capability of the past methods is limited when we only use the limited data to train. Now, we can look upon the traces of system call as consecutive logic actions. If we consider the short system call sequence is a computer action, an action space (or feature space) is made up of the actions. When the action space is partitioned into subspaces, the relationship of subspaces from the transition of short system calls sequences will be found. So we can use a little training data to achieve a concise model.

2.2 VQ

VQ has been used successfully in speech signal process and communication to compress information. Let T be the set of all possible vectors for the problem at hand. The task of VQ may be stated in the general case in which T is a continuous subset of R^k. T is separated into n distinct regions R_j that exhaust T, and each of them is represented with a $k-$dimensional vector, the so-call code vector or code word y_j, $j=1,\cdots,n$. $Y=\{y_i, i=1,\cdots,n\}$ is defined as the codebook. In the sequel, given a $x\in T$, we determine the region where it belongs, say R_j, and we adopt the corresponding representative y_j, instead of x. This is obviously associated with some information loss, which is known as distortion, $d(x,y_j)$. More details about VQ can refer to the literatures [18].

3. Anomaly detection method

3.1 notations

Let $\sum$ denote the set of system calls. A trace is simply a sequence of system calls generated by a process. A trace over $\sum$ is a finite sequence of system calls. The set of finite traces over $\sum$ is denoted by $\sum*$, and the set of traces of length n is denoted by $\sum_n$. Let O_{tr} denote the set of the training traces, $O_{tr}\subseteq\sum*$; O_{te} denote the set of the testing traces. Given a trace $\alpha\in\sum*$, $|\alpha|$ denotes the length of the trace; α_i and $\alpha[i]$ denote the prefix consisting of the first i system calls and the i th system calls respectively.

3.2 Feature Extraction

A sliding window $w(k)$ of size k is used to run across the sample $\alpha\in O_{tr}$, and move one symbol at one time:

$$x_i=\alpha(k+i)\cdot w(k) \tag{1}$$

We use SVQ algorithm to obtain a set of code word $y_i, i=1,\cdots,n$ as following:

Step 1. Initialization: cluster the entire training vector $X=\{x_i, i=1,2,\cdots\}$ into one vector codebook $Y=\{y_i, i=1\}$. The only code word is the centroid of the entire set of training vectors.

Step 2. Double the size of the codebook Y by splitting each code word in current Y according to the rule:

$$y_i^+ = y_i(1+\varepsilon)$$

$$y_i^- = y_i(1-\varepsilon),\ i = 1,2,\cdots,|Y| \qquad (2)$$

Where ε is the splitting parameter.

Step 3. Use the K-means iterative algorithm to get the best set of code words (centroids) for the splitted codebook.

Step 4. Iterate step 2 and 3 until the size of the codebook Y is equal to n which is a preestablished parameter.

Step 5. Use A_i to denote $y_i, y_i \in Y$.

Let A_i as the feature vector and construct a dictionary (or database) $N_A = \{A_1, \cdots, A_n\}$ of the unique feature vector of short sequences encountered in the normal sample α.

3.3 Constructing Markov Chains model of the feature vectors

Define A_i as the state of MC. Then the state space S_{MC} of MC is $S_{MC} \subseteq R^k$. The transition probability matrix P and the initial probability distribution π of a stationary MC can be learned from the observations of the system state in the past. In the paper, the frequencies of the states transition are used to approximate the transition probabilities. Provided with the observations of the system state $X_0,\ X_1, X_2, \ldots, X_{n-1}$ at time $t = 0, \cdots, n-1$, the transition probability matrix can be calculated by the following procedure.

$$p_{ij} = \frac{N_{ij}}{N_{i\cdot}},\ i \neq 0, j \neq 0 \text{ and } \sum_{j \neq 0} p_{ij} = 1 \qquad (3)$$

Where N_{ij} is the number of observation pairs X_t and X_{t+1} with X_t in state i and X_{t+1} in state j; N_i is the number of observation pairs X_t and X_{t+1} with X_t in state i and X_{t+1} in any one of the states.

By observation, it can be found out that the initial state is fixed if only the size k is selected correctly. Then the initial probability distribution is defined as $\pi = [1,0,\cdots,0]$.

3.4 Anomaly Detection

Anomaly detection matches current behavior against the normal behavior models and calculates the probability, which it is generated out of each model. The probability that the l consecutive states before time t occurs in the context of the stationary MC is computed as follows:

$$P_t(S_{t-l+1}, \cdots, S_t) = \pi_{S_{t-l+1}} \prod_{i=t-l+1}^{l} p_{S_{i-1}S_i} \qquad (4)$$

The higher probability can be got, the more likely the sequence of states results from normal activities. A sequence of states from intrusive activities is expected to receive a low probability of support from the Markov model of the norm profile.

4. Experiments

The system call dataset used in experiments have been obtained from the University of New Mexico. All of these data sets are publicly available and carefully described at http://www.cs.unm.edu/immsec. The 10% of normal data was used for training, and the remaining 90% of normal data as well as the intrusion data for testing.

For performance evaluation, two measures were defined, namely, true positive rate (TPR), false positive rate (FPR):

$$\text{TPR} = \frac{\text{number of intrusive testing traces detected as intrusive}}{\text{number of intrusive traces in testing set}}$$

$$\text{FPR} = \frac{\text{number of normal testing traces as intrusive}}{\text{number of normal trasces in testing set}}$$

These are good indicators of performance, since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications it makes in the process.

Table 1. The results when n vary

	n =64	n =128	n =256
TPR	100%	100%	100%
FPR	171/3000	21/3000	93/3000

From Table 1, it is found out that when $n=128$, the best performance can be obtained.

The comparison between the probabilities of the normal processes and those of the abnormal is shown in Table 2. There existed a clear gap between the minimum probability of the l consecutive states from the normal activities and the maximum probability of the l consecutive states from the intrusive activities. By using the middle probability value in the gap as the decision threshold to signal intrusions, the normal activities could be clearly distinguished from the intrusive activities. About parameter l, 6 is better than others.

Table 2. The result for testing when $n=128$

k	l	The $\min(P^l)$ of normal data	The $\max(P^l)$ of abnormal data	**TPR**	**FPR**
5	2	6.2981e-006	3.9943e-013	100%	37/3000
5	4	3.3392e-006	1.5954e-025	100%	33/3000
5	6	8.1144e-009	6.6476e-026	100%	21/3000
5	8	1.9719e-011	6.1612e-026	100%	207/3000
5	10	2.1199e-013	6.1612e-026	100%	151/3000
5	11	5.1704e-015	3.355e-026	100%	59/3000
5	12	5.1704e-015	3.1095e-026	100%	72/3000

5. Conclusion

A novel method for anomaly intrusion detection is proposed based on splitting vector quantization. The splitting vector quantization is employed to map short system calls sequences of the privileged processes to n disjoint feature subspaces. Markov chain is used to learn the relationship among these vector subspaces and classify the normal or abnormal. The observed behavior of the system is analyzed to infer the probability that the Markov chain of the norm profile supports the observed behavior. A low probability of support indicates an anomalous behavior that may result from intrusive activities.

For the intrusion detection involving system call sequences, the temporal dependencies are salient features. In this paper, the splitting vector quantization is used to extract and compress features. The primary experiments show that this method enjoys better generalization ability.

6. References

[1] M. Biswanath, T.Heberlein, and K. Levitt, "Network Intrusion Detection", IEEE Network, 8, pp. 26-41, May, 1994.

[2] Teng, H. S., Chen, K., and Lu, S. C., "Adaptive Real time Anomaly Detection Using Inductively Generated Sequential Patterns", Proceedings of the IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, 1990.

[3] Habra, N., Charlier, B. L., Mounji, A., and Mathieu,I., "ASAX: Software Architecture and Rule-based Language for Universal Audit Trail Analysis", Proceedings of the European Symposium on Research in Computer Security, Brighton, England, 1992.

[4] Helmer, G., Wong, J., Vasant Honavar, A., and Mille, L., "Intelligent Agents for Intrusion Detection and Countermeasures", Proceedings of the IEEE Information Technology Conference, Syracuse, NY, 1998.

[5] Lee, S. C. and Heinbuch, D. V., "Training a Neural network Based Intrusion Detector to Recognize Novel Attacks", Proceedings of the IEEE Workshop Information Assurance and Security, West Point, NY, 2000.

[6] Rhodes, B. C., Mahaffey, J. A., and Cannady, J. D., "Multiple Self-Organizing Maps for Intrusion Detection", Proceedings of the 23rd National Information Systems Security Conference, Baltimore, MD, 2000.

[7] Lin, T. Y., "Anomaly Detection - A Soft Computing Approach", New Security Paradigms Workshop, Little Compton, Rhode Island, 1994.

[8] Hiren Shah, Jeffrey Undercoffer, Dr. Anupam Joshi, "Fuzzy Clustering for Intrusion Detection," Proceedings of the 12th IEEE International Conference on Fuzzy Systems, April 2003

[9] S. Mukkamala, G. Janowski, A. H. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," Proceedings of IEEE IJCNN, pp. 1702-1707, 2002

[10] Wenjie Hu, Yihua Liao, and V. Rao Vemuri, "Robust Support Vector Machines for Anamoly Detection in Computer Security," International Conference on Machine Learning, Los Angeles, CA, July 2003

[11] Lee, W., Nimbalkar, R. A., Yee, K. K., Patil, S. B., Desai, P. H., Tran, T. T., and Stolfo, S. J., "A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions," in Recent Advances in Intrusion Detection (RAID 2000), Third International Workshop, Toulouse, France, October 2-4, 2000, vol. Vol. 1907, H.Debar, L. Me, and S. F. Wu, Eds. Berlin: Springer-Verlag, pp. 49-65,2000.

[12] Lee, W. and Stolfo, S. J., "A Framework for Constructing Features and Models for Intrusion Detection Systems," ACM Transactions on

Information and System Security, vol. 3, November, 2000
[13] Dit-Yan Yeung, Yuxin Ding, "Host-based intrusion detection using dynamic and static behavioral models," Pattern Recognition, 2003, 36(1): 229-243.
[14] Forrest, S., Hofmeyr, S. A., and Somayaji, A., "Computer Immunology," Communications of the ACM, vol. 40, pp. 88-96, 1997.
[15] Thottan, M. and Ji, C., "Proactive Anomaly Detection Using Distributed Intelligent Agents," IEEE Network, vol. 12, pp. 21-27, 1998.
[16] Lane, T. and Brodley, C. E., "Temporal sequence learning and data reduction for anomaly detection," ACM Transactions on Information and System Security, vol. 2,pp. 295-331, 1999.
[17] Kosoresow, A. P. and Hofmeyr, S. A., "Intrusion Detection via System Call Traces," IEEE Software, vol. 14, pp. 24-42, 1997.
[18] R.M.Gray. Vector Quantization. IEEE.ASSP Magazine, Volume 1:4-29,1984