AN ADAPTIVE KNN ALGORITHM FOR ANOMALY INTRUSION DETECTION

Assosiate profesor, PhD Veselina Jecheva, BFU
Assosiate profesor, PhD Evgeniya Nikolova, BFU

Abstract: *Intrusion detection is the process of monitoring the activity in the target system and analyzing it for intrusive activity. The purpose of the present paper is to create a methodology for the anomaly-based intrusion detection, which is grounded on the kd trees for the description of the system activity and the k-Nearest Neighbor algorithm during the intrusion detection phase, as well as the evaluations of the results.*

Keywords*: Intrusion Detection, Anomaly Based IDS, kd Trees, KNN algorithm, String Metrics.*

1. Introduction.

With the development and propagation of network technologies and applications, the malicious activities increased as well due to software bugs, hardware or software failures, or incorrect system administration. These issues have raised new concerns regarding security. Intrusion detection systems (IDS), as a second line of defense, are an unavoidable tool for detection of unauthorized access to some valuable system resources, violations to security policy and/or account theft. They monitor the system traffic for intrusive activity and in the case such activity occurs, they raise an alarm signal.

Axelsson identified two major types of intrusion detection strategies: anomaly detection and signature detection (misuse detection) [2]. Signature detection relies on the use of predetermined signatures of undesired behavior and can effectively detect known attacks and violations. Its major drawback is the inability to discover previously unknown intrusions.

The anomaly detection strategy is based on previously described normal system activity and flags everything that is unusual for the subject (computer, user, etc.) as suspicious. Traditional anomaly detection approaches build models of normal data and detect deviations from the normal model in observed data [15]. Therefore they have the ability to detect novel attacks or variations of already registered intrusions. Data mining is among the most frequently applied approaches in intrusion detection. It attempts to extract implicit, previously unknown and potentially useful information from data [4].

Anomaly detection has remained one of the most difficult tasks in data mining due to the inherent difficulty in precisely defining and quantifying the notion of anomaly and the big amount of data, that has to be processed. Anomaly detection has to be typically customized to the application domain, since its definition is domain-dependent [1].

Another important issue regarded to the anomaly detection approach is how to measure the deviation between the current activity and the normal activity profiles. Among the most applied methods is the application of the data mining techniques with the purpose to reduce and classify the observed data, as well as some string distances and similarity coefficients application in order to measure the degree of closeness between the current and normal activity data. A similar methodology is proposed by Liao and Vemuri [17], where the k-Nearest Neighbor (KNN) classifier was employed with an analogy between classifying text documents and detecting intrusion using the sequences of system calls. Another approach was proposed by Rawat, Gulati, Pujari and Vemuri [20], where each system call is treated as a word and cosine similarity measurement is used to calculate the distance between the current and normal behavior in order to detect the intrusions presence.

The present paper addresses the issue of the anomaly based IDS. The proposed methodology creates a normal activity database by analyzing data, which are collected by monitoring the behavior at the level of the privileged processes. The building of privileged program profiles has become a popular alternative to building user profiles in intrusion detection. These processes are frequent targets of the intruders, since they are granted more rights, compared to the ordinary users. Furthermore, their behavior is relatively stable over time. The normal activity database is composed using data mining techniques, i.e. classification trees, whose nodes consist of system calls sequences. The intrusion detection itself is performed using the KNN classification algorithm [25]. The intrusion detection algorithm is based on the KNN algorithm, which applies Jaro (JD) and Jaro-Winkler distances (JWD) as measures of the closeness of the current activity to the normal one.

2. Outline of the methodology.

The proposed methodology consists of two stages: the first one includes the normal activity description and the second one contains the intrusion detection itself. The normal activity data contains list of system calls, obtained by monitoring the behavior of some privileged processes for a certain period of time. This list is divided into sequences of system calls with length *L*, which are represented as a k-dimensional tree (kd-tree) [16], which is a space-partitioning data structure, usually applied for storing and clustering points, as well as fast binary search in a multidimensional space. The underlying space is decomposed into two halves each time a new point is inserted. All non-leaf nodes generate a splitting *k*-1 dimensional hyperplane, which divides the space into two subspaces [7]. The direction of hyperplane, i.e. the dimension on which the division is made, alternates between the *k* possibilities from one tree level to the next. Each splitting hyperplane contains at least one point, which is used as the hyperplanes representation tree [9].

The major purpose of the kd-trees is to hierarchically decompose the multidimensional space into a relatively small number of cells, which contain comparatively little number of the input objects. In our case the kd-tree nodes are sequences of normal system activity calls with length *L*. This representation provides a fast approach to access any input object by position. This data structure is very efficient, especially when the dimension of the data space is small, which is the examined case. When the number of dimensions increases then for very large databases its performance degrades exponentially and then it is suggested to use another structure [8].

The second stage consists of the intrusion detection process. In order to distinguish the normal activity patterns from the abnormal ones, a methodology based on the K-Nearest Neighbor algorithm (KNN) algorithm, was proposed. The purpose of this method is to classify objects based upon the closest training samples in the feature space [10]. KNN is a supervised learning algorithm where the membership of a new sample is classified into one of preliminarily specified classes based on majority of *K*-nearest neighbor category. The algorithm finds *K* number of objects or training points closest to the tested sample. The classification is performed using majority vote among the classification of the *K* nearest objects [21]. The closeness of the training and testing samples is calculated using the Jaro and Jaro-Winkler distances ([14], [24]).

3. Simulation experiments

The proposed methodology was tested by number of simulation experiments. The training and testing data were obtained by the Immune Systems Project, performed in the University of New Mexico [23]. These data were generated from Unix system examination

during specific period of time. The system behavior monitoring is performed by capturing system calls, made by some privileged processes under normal operational conditions. The input files contain normal user activity patterns of some privileged processes executed on behalf of the root account as well some anomalous data ([12], [13]). The input data files consist of sequences of ordered pairs of numbers, where the first number is the process ID (PID) of the executed process and the second one is the system call number.

As a first stage in our experiments, a set S_1 of unique short sequences by enumerating all unique, contiguous sequences of a normal user activity patterns with fixed length L was created. The elements of the set were represented as a kd-tree. An analogous procedure was performed with the data, containing anomalous patterns. As a result, a set S_2, which contains anomalous pattern sequences and a corresponding kd-tree was created. The testing data contain both normal and anomalous patterns for the following processes: xlock, login, named and synthetic sendmail. The simulation experiments were conducted by sliding a window of length L = 7, 10 and 13 across each test data set and adding each unique sequence to the database.

The second stage of the experiments includes the intrusion detection process. For that purpose the KNN algorithm was applies using *JD* and *JWD* with the following values of K=10, 20 and 30. The performance of KNN algorithm depends on the value of K. A large value of K makes boundaries between classes blur, while a small one may lead to a large variance in predictions [10]. Therefore, an appropriate value of K should be selected carefully in order to increase the accuracy and to reduce the false alarms. Typically the value of K is empirically determined and in our experiments it varies from 10 to 30 with step 10.

4. Evaluation of the results

The final stage is to assess the performance of the proposed classification method. Common method used in the machine learning and information retrieval community is: accuracy. For every possible criterion value there are four different possibilities to account for:

- True positive (*TP*), an attack correctly predicted;
- False positive (*FP*), an attack predicted when there was none;
- True negative (*TN*), no attack predicted when there was none to predict;
- False negative (*FN*), no attack predicted when there was one to predict.

Accuracy is the degree of correctness of such system [22].

$$Accuracy = \frac{number\ \ of\ \ TP\ \ +\ \ number\ \ of\ \ TN}{numbers\ \ of\ \ TP + TN + FP + FN}$$

An accuracy of 100% means that the test identifies all anomalous and normal activity correctly. If an IDS raises an alarm for the legitimate activity of a user, then a false alarm is present. An intrusion detection system becomes more accurate as it detects more attacks and raises fewer false alarms. The conducted experiments indicate that the proposed methodology produces results with high level of accuracy, since all obtained values are between 81,45% and 98,44% for all examined processes. Table 1 contains the accuracy values for all examined processes with respect to the various values of K in the case when L=7. At a false positive rate of 5%, the proposed method outperforms the results obtained by [17] with average detection rate of 95%. Compared the algorithm performance to the results obtained by Rawat et. al. [20], we can conclude that their methodology yields better accuracy than ours, but our algorithm achieves lower average false alarms rate.

Table 1. The accuracy values depending on K when $L=7$

Processes	Distance	*K*=10	*K*=20	*K*=30
synthetic sendmail	*JD*	81,45%	92,45%	91,54%
	JWD	81,84%	90,46%	91,30%
login	*JD*	95,34%	97,74%	98,44%
	JWD	95,47%	97,26%	97,78%
named	*JD*	91,08%	97,89%	97,97%
	JWD	90,65%	97,38%	97,50%
xlock	*JD*	85,04%	92,65%	96,98%
	JWD	86,43%	93,45%	95,93%

The results obtained for *L*=10 and 13 are shown in Tables 2. From table it is easily observed that the detection accuracy of the proposed method is sufficiently large for *K*=30 and both values of *L*, since all values are greater than 86%. The minimum accuracy (between 86% and 87%) is achieved for *K*=10 and *L*=13. All other values are greater than 93%, which means the proposed methodology yields excellent accuracy results. It can be seen that the results are variable with different *L*, which means that the performance of the intrusion detection depends on the sequence length. Therefore, the choice of the value of *L* is meaningful.

Table 2 *Accuracy values, when L=10 and 13*

		L=10			*L*=13		
Process	Distance	*k*=10	*k*=20	*k*=30	*k*=10	*k*=20	*k*=30
login	*JD*	93,20%	99,18%	98,93%	86,71%	93,42%	99,04%
	JWD	93,24%	97,82%	99,36%	86,32%	94,13%	98,06%

Conclusion

Supervised network intrusion detection has been an area of active research for many years. The present paper proposes an anomaly-based approach, which applies some data mining techniques for the normal data description and the kNN algorithm during the detection phase. The experimental results with a host-based dataset demonstrate that the proposed method is robust and effective while detecting the violations of the computer security.

References:

1. Agovic A., A. Banerjee, A. R. Ganguly, V. Protopopescu, Anomaly Detection in Transportation Corridors Using Manifold Embedding, *Knowledge Discovery from Sensor Data*, CRC Press, 2008, pp. 81-105.
2. Axelsson S., The base-rate fallacy and the difficulty of intrusion detection, *ACM Transactions on Information and System Security (TISSEC)*, Volume 3, Issue 3, August 2000, pp. 186 – 205.
3. Baldi P., Brunak S., Chauvin Y., Andersen CAF, Nielsen H., Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 16, 2000, pp. 412–424.
4. Barbara D., S. Jajodia, *Applications of Data Mining in Computer Security*, Springer, 2002, CRC Press, 2008, pp. 81-105.
5. Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Ser. B, 57, 289–300.

6. Benjamini,Y. and Hochberg,Y. (2000) On the adaptive control of the false discovery rate in multiple testing with independent statistics. J. Edu. Behav. Stat., 25, 60–83.
7. Bentley J.L., Multidimensional Binary Search Trees Used for Associative Searching, Communications of the ACM, Vol.18, Num. 9, 1975.
8. Berchold S., Kriegel H. P., Indexing the Solution Space: A New Technique for Nearest Neighbor Search in High-Dimensional Space, *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, No. 1, 2000, pp. 45-57.
9. Chmielewski A., S. T. Wierzchon, V-Detector algorithm with tree-based structures, *Proceedings of the International Multiconference on Computer Science and Information Technology*, Wisla, Poland, 2006, pp. 11-16.
10. Dasarathy B.V., Nearest Neighbor (NN) norms: NN Pattern Classification Techniques. IEEE Computer Society Press, 1990.
11. Ferri C., N. Lachinche, S. A. Macskassy, A. Rakotomamonjy, eds. (2005). Second Workshop on ROC Analysis in ML.
12. Forrest S., S.A. Hofmeyr, A. Somayaji, T.A. Longtaff, A sense of self for Unix processes, In Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitors, CA, pp.120-128.
13. Forrest S., S.A. Hofmeyr, A. Somayaji, Intrusion detection using sequences of system calls, *Journal of Computer Security*, Vol. 6, 1998, pp. 151-180.
14. Jaro M. A., Advances in record linking methodology as applied to the 1985 census of Tampa Florida, *Journal of the American Statistical Society*, 1989, 414-420.
15. Lazarevic A., A. Ozgur, L. Ertoz, J. Srivastava, V. Kumar, A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection, *Proceedings of the third SIAM International Conference on Data Mining*, 2003, pp. 25-36.
16. Lee D. T.; Wong, C. K., Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees, Acta Informatica, Vol. 9, Num. 1, 1977, pp. 23–29.
17. Liao.Y and Vemuri.V.R, "Use of k-nearest neighbour classifier for intrusion detection", *Int. J. of Computer Security*, Vol.21, No.5, Oct 2002, PP.439-448.
18. Lippmann R., D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Webber, S. Webster, D. Wyschograd, R. Cunninghan, M. Zissan, "Evaluating Intrusion Detection Systems: the 1998 DARPA off-line Intrusion Detection Evaluation", Proceedings of the DARPA Information Survivability Conference and Exposition, IEEE Computer Society Press, Los Alamitos, CA, 12-26, 2000.
19. Matthews, B.W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim*. Biophys. Acta, Vol. 405, pp. 442-451.
20. Rawat S., V. P. Gulati, Arun K. Pujari, V. Rao Vemuri, Intrusion Detection Using Text Processing Techniques with a Binary-Weighted Cosine Metric, Journal of Information Assurance and Security 1 (2006) 43–50.
21. Shakhnarovich G., T. Darrell, P. Indyk, Nearest-neighbor Methods in Learning and Vision: Theory and Practice, MIT Press, 2006.
22. Taylor J. R., An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. University Science Books, 1999, pp.128-129.
23. University of New Mexico's Computer Immune Systems Project, http://www.cs.unm.edu/~immsec/systemcalls.htm
24. Winkler W. E., The state of record linkage and current research problems, *Statistics of Income Division*, Internal Revenue Service Publication R99/04, 1999.
25. Yang Y., An Evaluation of Statistical Approaches to Text Categorization, Technical Report CMU-CS-97-127, Computer Science Department, Carnegie Mellon University, 1997.