Constructing Cost-effective Anomaly Detection Systems for User Identification

Charlie Y. Shim, and Jung Yeop Kim, *Member, IAENG*

***Abstract*— User based anomaly detection systems identify intrusions by creating normal profiles for a user and comparing the current pattern with his/her profiles. We propose to apply the SVM algorithm to the concurrently employed sequence of user commands that have been weighted according to their frequencies to identify system users. Our approach is not only simple but also robust in controlling noisy data.**

***Index Terms*— Intrusion detection systems, misuse/anomaly detection, normal profiles, support vector machines**

I. Introduction

The need to devise mechanisms for maintaining comprehensive security of computer systems cannot be emphasized enough. Novel attacks are being developed continually, making it hard for systems to be made immune to all vulnerabilities. Among the mechanisms that are used to protect systems against unauthorized activities is to attempt to detect intrusions; if done early enough, intrusion detection can help minimize losses from improper or destructive uses [1]. Therefore, Intrusion Detection Systems (IDSs) have become a crucial part of network security metrics for over twenty years; moreover, evaluations that focus on intrusion detection algorithm/performance are essential for ongoing research [2].

IDSs utilize one of two mechanisms: Misuse and Anomaly detection. Misuse detection mechanisms define a set of "unacceptable" activities and raise alerts when behavior matches this set [3, 4, 5]. On the other hand, Anomaly detection mechanisms create a profile of typical behavior of the system/user and raise an alert when the activity does not fit its normal profile [6, 7, 8]. We are particularly interested in constructing efficient anomaly detection systems for effective user identification.

There are three issues that need to be addressed when describing normal behavior. The first issue is how to determine what information constitutes a good database of normal patterns. The Longman Dictionary of Contemporary English defines "normal" as "not unusual in any way, but happening just as you would expect" [9], so we consider normal behavior as usual patterns that have been observed. However, there is no explicit or standardized method that can fully describe normal behavior; thus, intelligent profile construction schemes to characterize normal behavior critically affect the overall performance of the IDSs. Dissimilar information constitutes normal profiles differently depending on the type of intrusion detection system.

The second issue in creating normal profiles is how to reduce the dramatic dimensionality and complexity of the data. A common challenge in intrusion detection is that a high volume of information needs to be accumulated and stored in order to match input traces against the data. The question is how to create profiles of a user's behavior without having to slow down the system's performance due to the data issue. Existing approaches attempt to wait until sufficient profiles are collected in order to retrieve all possible patterns; thus the data amount for profiling can easily become extremely large. However, excessive simplification of data can lead to false positive/negative alarms, which leads to tradeoffs between computation cost and reliability. Including too much data will adversely impact the performance of the system, whereas considering too little data will reduce the overall effectiveness [10]. The problem arises when large data storage is required due to inefficient data processing procedures. It then becomes a huge burden to manage all the details of such information.

The third issue, related to the previous problem, is how to control the noise effect when reducing data dimensionality. Normal profiling is a delicate and complex process, and again there is a tradeoff between data dimensionality and the noise effect that is caused by unusual data entry executed by users. If we keep the amount of normal profile to a minimum, data dimensionality will be significantly reduced; however, it is likely to be contaminated much more by the uncommon patterns, which need to be controlled. For instance, it is quite natural and likely that users may temporarily stop working and check emails; however, this action will cause a mismatch between the normal behavior of the user and audited patterns. Classification of detection will produce false alarms if testing data is directly compared with normal profile whose instantaneous patterns have not been previously filtered. For this reason, a mechanism to control variations in the user's temporal patterns is necessary.

II. Related Work

Selecting an appropriate set of features for normal profiles is directly related to achieving efficient anomaly detection systems in that data dimensionality in the system can be reduced. However, even though the best set of features can

Manuscript received July 27, 2008.

Charlie Y. Shim is with the Department of Computer Science, Kutztown University of Pennsylvania, Kutztown, PA 19530 USA. (e-mail: shim@kutztown.edu).

Jung Yeop Kim is with Department of Computer Science, Utica College, Utica, NY 13502 USA. (e-mail: jkim@utica.edu).

be selected, the dimensionality of data will be dramatically increased as the size of profile data increases.

The usual approach to alleviate this burden is to cluster the profile data – that is, partition the data into meaningful subgroups based on some aspects of similarity measure. Portnoy et al. used a clustering algorithm to group unlabeled data, while assuming that the proportion of anomalous data was low [11]. A cluster is a collection of data objects that are similar to one another within the same cluster but are dissimilar to the objects in other clusters [12]. Since clustering deals with finding unlabeled collection of data, applying a clustering algorithm to anomaly detection systems is a good idea, as anomalous activities cannot be labeled due to their unpredictable nature. However, a generic type of clustering algorithms such as k-means clustering requires the size of clusters to be defined in advance. This is difficult because it will affect the overall performance of the classification. There is a tradeoff between reliability and efficiency - that is, a large value of k results in an expensive computation, while reliability is degraded when k becomes too small.

In order to overcome this dilemma, Burbeck et al. proposed ADWICE (Anomaly Detection with Fast Incremental Clustering) to detect anomalies in network data without a pre-determination of the cluster size, since only compact summaries of clusters were kept in memory rather than the complete data set [13]. Although such an approach reduces data dimensionality by grouping similar features together, applying a clustering algorithm is not immune to data contamination due to irrelevant information ("noise") that needs to be controlled.

A response to the noise problem is found in Lane et al., who collected temporal sequences of UNIX user commands and differentiated the profiled user from masqueraders by a characterization of valid user behavior [14]. They identified intrusion attempts by investigating similarity between testing commands and a user's normal profile. Their main idea was to assign a greater weight to adjacent matches. Since an irrelevant coincidence is likely to create contamination in the normal profile, they applied a noise-suppression filter to the resulting data stream that was previously compared to the user's historical profile so that the classification of the data stream could be smoothed [14]. One drawback of their approach is that the complexity of the system has been increased due to the installation of the extra filter.

III. Cost-Effective Anomaly Detection

What our research addresses is to construct efficient anomaly detection systems so that user identification can be effectively handled. This is accomplished in two steps. First, our approach to characterizing normal behavior is based on creating sequences of concurrent UNIX user commands employed by each user. Then, the problem of data dimensionality vs. noise effect is controlled using a classification algorithm called Support Vector Machine (SVM).

Our method investigates the sequence of concurrent user commands for the purpose of identifying the patterns of the user. It is based on the idea that a command sequence represents each user's behavior over a given period of time uniquely, because users usually perform routine tasks daily unless their assigned jobs significantly change; thus, they tend to use the same patterns of commands.

We document the concurrently employed user command sequences and assign proportional weights to the repeatedly used patterns. The core concept of profiling is to recognize the normal patterns of the user and the frequency constitutes the measure of normality. Since concurrent employment of the same commands is a good indicator of identifying each user, we investigate the regularity by assigning proportional weights to the frequently employed concurrent commands. Then, a normal profile is a set of instances where a single instance is a fixed length command sequence. Given the sequences of UNIX commands, we parse and partition the sequences into meaningful subgroups of fixed length that constitute a set of normal user profiles.

If the number of command items in a single instance is denoted by the letter "μ", then a single instance of command sequence is the concatenation of existing (μ-1) sequences and the most current command input. This method of sequence partitioning has been used by many other researchers [14, 15]. Accumulating the previous (μ-1) sequences enhances the preciseness in comparing the patterns between commands. When the letter "i" represents the index of normal instances, a normal profile for each user can be represented as a set of instances Rs = {ri | i Znonneg} where ri is a single instance.

There is one question that needs to be answered, however. How many UNIX commands need to be recognized and processed? Some researchers used all available commands for the purpose of achieving high classification accuracy [10]; however, when this was done, the actual performance of the detection rate worsened. This result can be inferred from the fact that all possible commands (there are almost 1100 different commands) were analyzed using a relatively small training dataset.

Then, we apply a SVM, a class of well-founded mathematical algorithms, to the set of instances from which we extract the normal patterns of each user after partitioning the sequence of UNIX commands into meaningful subgroups. The SVM algorithm is more robust to the noise effect since it finds a better classifier by finding a maximum margin hyper plane between two classes of target concepts. Controlling the noisy data is important since "overfitting" phenomenon due to irrelevant examples in training data causes the learner to perform well on the training dataset while increasing validation errors.

IV. Conclusion

There are more details that need to be addressed to make our approach functional. For instance, as mentioned in the previous section, we need to select key command items that can determine the most informative profile features for the system. Also, we need to identify the ideal length of μ that is best suited to represent the characteristics of each user. Furthermore, there are many SVM applications available and we must evaluate which SVM is most appropriate for our purpose.

However, applying a SVM algorithm to the concurrently employed sequence of user commands that have been weighted according to their frequencies is a simple but powerful approach for anomaly detection and user identification. Our approach does not require additional noise reduction process and therefore reduces the complexity of the system, which is a step towards making computer systems more cost-effective.

REFERENCES

[1] Gaurav Tandon and Philip Chan, "Learning Rules from System Call Arguments and Sequences for Anomaly Detection," ICDM Workshop on Data Mining for Computer Security (DMSEC), pp. 20~29, 2003.

[2] J. Qian, C. Xu, and M. L. Shi, "Remodeling and Simulation of Intrusion Detection Evaluation Dataset," Proceedings of the 2006 International Conference on Security and Management (SAM), June 2006.

[3] [3] Phillip Porras, and Richard Kemmerer, "Penetration State Transition Analysis: A Rule-Based Intrusion Detection Approach," Proceedings of the Eighth Annual Computer Security Applications Conference, December 1992.

[4] Sandeep Kumar, and Eugene Spafford, "A Pattern Matching Model for Misuse Intrusion Detection," Proceedings of the 17th National Computer Security Conference, pp. 11~21, 1994.

[5] Giovanni Vigna, and Richard Kemmerer, "NetSTAT: A Network-Based Intrusion Detection Approach," Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC), pp. 25~34, December 1998.

[6] C. Ko, M. Ruschitzka, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-based Approach," Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 175~187, May 1997.

[7] A.K. Ghosh, J. Wanken, and F. Charron, "Detecting Anomalous and Unknown Intrusions Against Programs," Proceedings of the Annual Computer Security Applications Conference (ACSAC), pp. 259~267, December 1998.

[8] David Endler, "Intrusion Detection Applying Machine Learning to Solaris Audit Data," Proceedings of the 1998 Annual Computer Security Applications Conference (ACSAC), 1998.

[9] Della Summers et al, Longman Dictionary of Contemporary English, Longman Corpus Network, 1995.

[10] John Marin, Daniel Ragsdale, and John Surdu, "A Hybrid Approach to the Profile Creation and Intrusion Detection," Proc. of DARPA Information Survivability Conference and Exposition, Information Technology and Operations Center, United States Military Academy, pp. 12~14, 2001.

[11] L. Portnoy et al, "Intrusion detection with unlabeled data using clustering," ACM Workshop on Data Mining Applied to Security (DMSA), November 2001.

[12] Jiawei Han, and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, p 336, 2001.

[13] Kalle Burbeck, and Simin Nadjm-Tehrani, ADWICE – Anomaly Detection with Real-time Incremental Clustering, Springer Berlin/Heidelberg, pp. 407~424, 2005.

[14] Terran Lane, and Carla Brodley, "Temporal sequence learning and data reduction for anomaly detection," ACM Transactions on Information and System Security, vol. 2, no. 3, pp. 295~331, 1999.

[15] Karlton Sequeira and Mohammed Zaki, "ADMIT: anomaly-based data mining for intrusions," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 386~395, 2002.