Abnormal Behavior Analysis in Office Automation System within Organizations

Yilin Wang, Yun Zhou, Cheng Zhu*, Xianqiang Zhu, Weiming Zhang
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China.

* Corresponding author. Email: zhucheng@nudt.edu.cn
Manuscript submitted June 10, 2017; accepted September 2, 2017.
doi: 10.177606/ijcce.6.3.212-220

Abstract: Insider threat is a serious and increasing concern for many organizations. The group of individuals who operate within the organization have access to highly confidential and sensitive information, however, if they choose to act against the organization, with their privileged access authority and their extensive knowledge, they are well positioned to cause serious damage. Compared with vast amounts of normal daily operations, malicious behaviors are indeed small probability events, and are easily ignored. Thus, there is a desperate need to explore an effective approach to detect such suspicious behaviors. In order to solve this problem, we propose a two-stage algorithm to detect anomaly through analyzing user behavior based on activity log data collected in a real office automation system. In the first stage, we compare users' behavioral activities with activities of his/her belonging role, and in the second stage, we compare individual behavioral activities with his/her activities in a window period. By adopting several effective features to describe users' regular behavioral patterns, the analyst is capable of refining underlying abnormal users and abnormal periods to better support the network security administration.

Key words: Cyber security, behavior analysis, anomaly detection.

1. Introduction

Enterprise network security management has been a problem for many years. Office automation system is applied to enterprises' daily operation and management and it plays an important role in the construction of enterprise information. The safety and stability of office automation system count a lot in constructing secure enterprise network. Traditionally, cyber threats come from the outside of the enterprise, however, more and more case studies show that threats not only originate from outside, but also inside the enterprise. According to the Small Business Administration in Korea, 90% of the information leakage incidents are made by internal staff [1]. Malicious insider threat has devastating impact on enterprises, because individuals who operate within the enterprise have access to highly confidential and sensitive information and if they act maliciously, it may pose financial and reputational damage to the enterprise. Since the damage is getting serious with the information leakage incident, insider security study has become one of the biggest issues in the cyber security.

Besides, monitoring and detecting abnormal activities can function in avoiding ineffectively occupying resources. Of course, abnormal behaviors may not always be threatening, however, malicious behaviors certainly behave abnormally. Therefore, there is a desperate need to explore an effective approach to detect suspicious behavior, and remind the network administrator of anomaly to mitigate the risk. However, there

are plenty of challenges in identifying abnormal behaviors, like how to define abnormal behaviors based on users' previous activities and how to measure anomaly scores, besides, how to detect anomaly at minimal cost and reduce negative impact on enterprise daily operation.

In this work, we propose a two-stage algorithm for analyzing behavioral activities and detecting abnormal behaviors. The algorithm presents an unified framework to detect both role-based and individual-based abnormal behaviors. In detail, we compare user's behavioral activities with activities of his/her belonging role, and compare individual behavioral activities with his/her activities in a window period. We calculate a score for each user and the user whose score exceeds a particular threshold is considered as anomaly. We can also find the periods when the user behaves differently comparing to his/her history. Besides, our data of experiment are collected in a real office automation system within an organization and we propose a number of effective features for profiling users' temporal activities and derive anomaly metrics to measure the degree of anomaly.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we present the two-stage algorithm combining Local Outlier Factor (LOF) and Isolation Forest methods for users' behaviors analysis. In Section 4, we analyze users' behaviors based on our collected dataset. In Section 5, we conclude the paper and discuss our future work in this field.

2. Related Work

Insider attacks have become the primary threat to the computer systems [2]. The topic of insider threat detection has received many attentions in the literature. Traditional intrusion detection system is neither designed for nor capable of identifying those who act maliciously within an organization [3]. Researchers have proposed a number of systems and approaches to detect or predict insider threat based on different types of activities. Myers *et al.* [4] used web server log files for insider threat detection. Eldardiry *et al.* [5] proposed a novel insider threat approach by combining several behavioral activity domains, and find discrepancies among these domains.

There are several studies on analyzing user behavior based on role and individual profiles. Anderson *et al*. [6] designed a behavior-anomaly-based system using peer-group profiling and monitored real-time process. Nguyen *et al*. [7] built a system to detect insider misbehaviors by monitoring system call activities, including file access and process execution. Moreover, in the user-oriented model, which they developed for analyzing file access, they considered building a profile for each user to distinguish misbehaviors from normal behaviors.

Philip A. Legg *et al*. [8] developed the Corporate Insider Threat Detection (CITD) system, which measured how user deviated from his previous observations and previous observations of his role to assess the potential threats. In addition, the feature set they considered includes the device that captured the log, the activities observed on the device, and the primary attributes associated with the activities. The feature set consists of three categories: users' daily observations, comparison- s between users' daily activities and previous activities of their roles, and comparisons between users' daily activities and their previous activities. However, they didn't consider temporal behavioral features, and they didn't compare user daily observations in a window period. Park etc. [9] developed the monitoring mechanisms with the role-based profile and individual-based profiles. Moreover, they used frequencies of corresponding events as metrics.

Besides, there exist work studying the characteristic of insider threat and evaluating the level of threat. Senator *et al.* [10] developed multiple algorithms for anomaly detection and demonstrated the feasibility of proposed methods for insider threat detection. Magklaras and Furnell [11] proposed a threat evaluation system based on profiles of user behaviors to estimate the level of threat.

There are a number of methods of anomaly detection, such as probabilistic and statistical models, linear

models, proximity-based anomaly detection, time series and multidimensional streaming anomaly detection [12]. Besides, there are some traditional approaches like supervised learning, which detects minority class by building data classification models from training data. Eberle *et al.* [13] considered graph-based anomaly detection as a tool. A study conducted by Parveen *et al.* [14] applied multiple methods, such as ensemble-based stream mining, unsupervised learning, and graph-based anomaly detection to insider threat detection.

3. A Two-Stage Algorithm for Anomaly Detection

We propose a two-stage algorithm for anomaly detection. First, we construct a feature set to classify users into roles, define anomaly metrics based on the feature set, and compare users' behavioral activities with activities of his/her belonging roles. Second, we compare individual behavioral activities with his/her activities in a window period. In the following sections, we present how each part of the algorithm is performed to identify abnormal behaviors.

3.1. The Framework of the Two-Stage Algorithm

In the first stage, we compare users' behavioral activities with activities of his/her belonging role. Based on users' activity data, a feature set is selected based on the result of the Random Forest classifier. Every user is examined by the LOF detector to judge whether the user is an anomaly or not, through comparing his/her LOF score with the role- based threshold. The goal of the first stage is to find abnormal users. In the second stage, we compare individual behavioral activities with his/her activities in a window period. The user whose LOF score exceeds the role-based threshold in the first stage is further examined by the Isolation Forest detector to find his/her abnormal periods, which are given with anomaly scores calculated by the Isolation Forest method. Periods which score above 0.5 are regarded as anomalies according to the discussion in [15]. The framework of this two-stage algorithm is presented in Fig. 1.

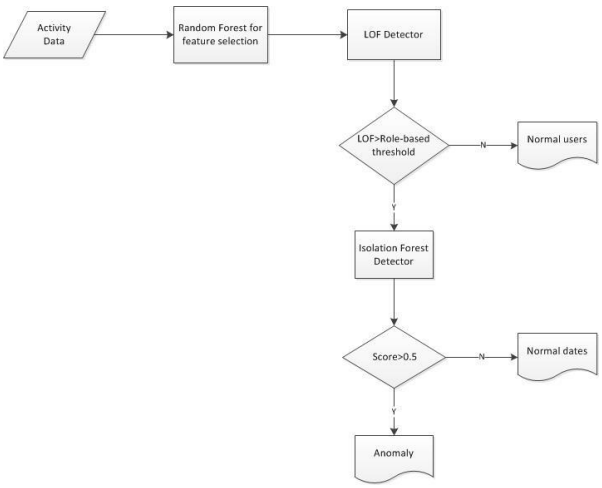


Fig. 1. The proposed two-stage algorithm.

3.2. The Combination of Random Forest and LOF in the First Stage

Let $D_N = \{(x_i, y_i)\}_{i=1}^{N}$ denote the dataset including N instances in the role-based profile. Each instance consists of d dimension features denoted as $x_i = (x_{i1}, x_{i2}, ..., x_{id})$, and each instance belongs to a role, which is formulated as y_i.

There are many combinations of features that can be extracted from raw data, however, features selected

manually may be irrelevant and redundant. Feature selection can function in refining effective features. Random Forest can be used to rank the importance of features, and at first, we fit a random forest to the data. The out-of-bag error can be used to measure the prediction error of the random forest due to bootstrapping. Out-of-bag error is computed for each tree as e_1, and after perturbing the data, the out-of-bag error is calculated as e_2. We calculate the degree of importance of a feature by averaging the difference between e_1 and e_2. A feature is more important if it produces higher error after perturbing the data. By imposing Random Forest on users, we construct an effective feature set.

We calculate anomaly score for each instance using LOF [16] as: $f_{LOF}(x_i)=\{l_1,l_2,...,l_N\}$. The LOF is used for detecting density-based outliers. Through comparing the density of the instance and densities of its neighbors of the same role, the instances that have obviously lower densities than their neighbors of the same role are regarded as outliers. LOF has been demonstrated as an effective method that can identify meaningful local outliers that previous approaches can not find [16].

Let $d(x_i,x_j)$ denote the Euclidean distance between x_i and x_j, $d_k(x_i)$ denote the distance from x_i to the k-th nearest neighbors of the same role, where these k-nearest neighbors $N_k(x_i)$ includes all instances within this distance. $|N_k(x_i)|$ denotes the number of k nearest neighbors of the same role. Given the parameter k, $reach-d_k(x_i,x_j)$ denotes the higher one between $d(x_i,x_j)$ and $d_k(x_i)$.

The local reachability density of x_i is the inverse of average reachable distance of instance x_i from its neighbors of the same role and is defined as:

$$lrd(x_i)=1/\left(\frac{\sum_{x_j\in N_k(x_i)} reach-d_k(x_i,x_j)}{|N_k(x_i)|}\right)$$

We calculate the LOF score for each instance by comparing local reachability of it and its neighbors of the same role:

$$f_{LOF}(x_i)=\frac{\sum_{x_j\in N_k(x_i)}\frac{lrd(x_j)}{lrd(x_i)}}{|N_k(x_i)|}$$

The calculated LOF equal to 1 means the instance has similar level of density with its neighbors of the same role, and is regarded as normal one, and when the calculated LOF score is significantly larger than 1 means the instance is an outlier. We compare LOF score of each instance with role- based threshold and apply Boxplot method to find anomalies, whose LOF scores exceed Q3 (upper quartile) + 1.5*IQR (interquatile range). Therefore, we find abnormal users.

3.3. The Isolation Forest Method for Periods Detection in the Second Stage

Let $R_T=\{t_i\}_{t=1}^{T}$ denote the dataset including T instances in the individual-based profile. Each instance consists of m dimension features denoted as $t_i=(t_{i1},t_{i2},...,t_{im})$ and each instance represents an user's behavior activities in a specific period. Isolation Forest is an effective and efficient method to detect users' deviation from normal behaviors in a window period. Isolation Forest is an algorithm with a low linear time complexity and a small memory requirement [15]. We calculate anomaly score for each instance as $f_S(t_i)=\{s_1,s_2,...,s_T\}$ using Isolation Forest.

An Isolation Forest (iForest) consists of plenty of isolation trees, which are built by selecting attributes and the values of attributes randomly. Instances are partitioned into two parts based on selected attributes and their values at each node of trees. Anomalies are the instances that require less partitions to be isolated. Given a dataset of T instances, path length $pl(t_i)$ is used to measure the degree of isolation and shorter path length means more likely to be anomalies. The average path length is defined as:

$$c(T) = \begin{cases} 2H(T-1) - 2(T-1)/n & for\ T > 2 \\ 1 & for\ T = 2 \\ 0 & otherwise \end{cases}$$

where $H(x)$ is the harmonic number. The anomaly score of an instance t_i is defined as:

$$f_S(t_i) = 2^{-\frac{E(pl(t_i))}{c(T)}}$$

where $E(pl(t_i))$ is the average of $pl(t_i)$. Using the anomaly score $f_S(t_i)$, we can conclude that normal instances have scores smaller than 0.5 and anomalies have scores close to 1. Besides, the dataset do not have obvious anomaly if all the instances have scores approximately equal to 0.5. We calculate scores for the user's activities in a window period, therefore, we find abnormal periods of abnormal users.

4. Experiment

4.1. Dataset

The data of experiment are collected within a real enterprise organization, which records users' operations in office automation system. There are 500 users in the enterprise, and we choose 3 representative roles and we mark them as y_1, y_2, y_3, among which y_1 represents senior manager, y_2 represents department manager, and y_3 represents staff. We filter inactive users in the system, therefore, the number of these 3 roles are respectively 20, 36, 142. The data for experiment contain activity data during a period of 228 days, from March 1st to October 10th. In addition, these real data are transformed after preprocessing, denoising and filtering sensitive information. Document operations are operations that are related to documents, which are recorded in log files, and attributes of each record include: doc ID, subject of document, operational type, operator ID, operational time, etc.

4.2. Feature Selection and Role-Based Anomaly Detection

4.2.1. Feature selection

Before we measure anomaly scores, we perform feature selection by Random Forest. Our features include three categories: operational type, operational object and time attribute. There are two operational objects: document and workflow. Document includes news, notification and process attached documents. Workflow describes a working process of requesting and replying.

The operational types of documents includes creating and reading, and workflow includes requesting and replying. Besides, we consider time attribute, which consists of two components, one is the ratio of user's behavioral activities that happened in a time period of 18:00 to 24:00 or on weekends (Sunday and Saturday) or in the morning (from 7:00 to 12:00) and in the afternoon (from 13:00 to 18:00), another is the total amount of user behavioral activities in a window period. Thus, our features are the combination of operational type, operational object, and time attribute. For example, the number of notifications (object)

that are read (type) that happened in a window period (time attribute). According to users' position within the organization, we divide users into three roles and we mark them as y_1, y_2, y_3.

We apply Random Forest classifier to classifying users into roles. Below we have extracted 9 types of features to characterize behaviors of a specific user :

N_1 : the number of notifications that a user has created;

N_2 : the number of notifications that a user has read;

N_3 : the number of attached documents that a user has read;

N_4 : the number of workflow that a user has requested and replied;

R_1 : the ratio of notifications that a user has read between morning and afternoon;

R_2 : the ratio of attached documents that a user has read between the period of 18:00-24:00 and the whole day;

R_3 : the ratio of attached documents that a user has read between weekend and the whole week;

R_4 : the ratio of workflow that a user has requested and replied between the period of 18:00-24:00 and the whole day;

R_5 : the ratio of workflow that a user has requested and replied between weekend and the whole week;

Degree of contribution of features are calculated using Random Forest, and the results are shown in Table 1. The higher the degree, the more important the feature.

Table 1. The Degree of Contribution of Features

N_1	N_2	N_3	N_4	R_1	R_2	R_3	R_4	R_5
0.08	0.24	0.08	0.24	0.06	0.11	0.13	0.13	0.08

4.2.2. Role-based anomaly detection:

We get distance matrix through calculating Euclidean Distance between each pair of users, and then we apply LOF to detecting role-based anomaly. The higher LOF score is, the more abnormal it is. Next, we rank users' LOF scores by a descending order and use the Boxplot method to find anomalies, and finally we get anomaly threshold for each role as 2.733, 1.8125, 1.7625. Fig. 2 presents anomaly situation of each role, *x* axis stands for the index of each user and y axis stands for LOF scores. There are several abnormal users of role y3 are detected in Fig. 2.

We take the user who scores the highest as an example and we find that he behaves more abnormally on multiple anomaly metrics (features N_2 and N_3).

4.3. Individual-Based Anomaly Detection

We compare individual behavioral activities with his/her activities in a window period. When profiling individual behavioral activities, we consider user's temporal activities for each day. And we select several representative activities including requesting and replying workflow, reading notifications, and reading attached documents. We partition each day into 24 time bins (i.e. a time bin lasts for 1 hour) and compare each day's activities at the same time bin.

Below we have selected adaptable anomaly metrics:

N_w : the number of workflow that a user has requested and replied.

N_n : the number of notifications that a user has read.

N_d : the number of attached documents that a user has read

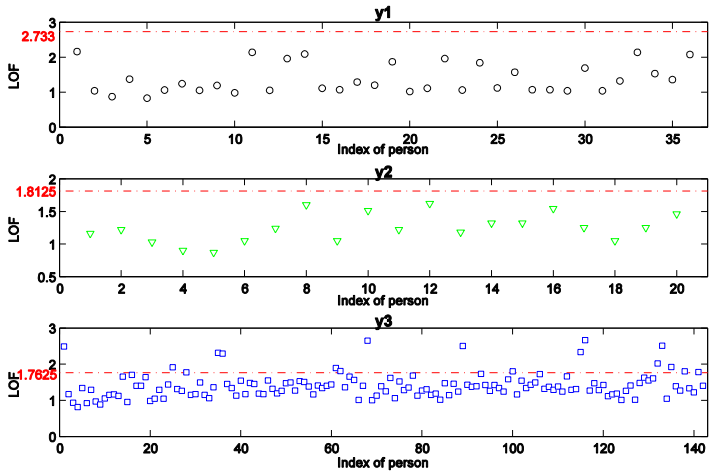


Fig. 2. LOF scores and thresholds for each role.

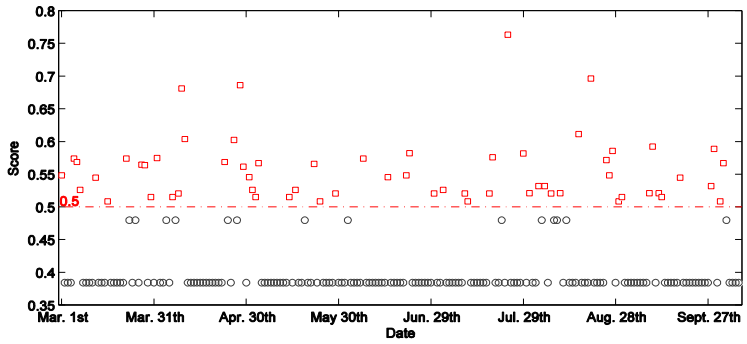


Fig. 3. Anomaly score of the representative user for the 9th time bin of each day.

We apply Isolated Forest to detecting abnormal dates for abnormal users. In Fig. 3 we present an example which is scores of 9th time bin each day of the representative user who scores the highest in stage one. We detect several abnormal dates which are marked as red squares in Fig. 3.

5. Conclusion

In this paper, we present a two-stage algorithm to detect abnormal behavioral activities. The algorithm has been applied to analyzing the profile of all users who have access to office automation system. By incorporating role and individual based profile, the system is capable of obtaining a comprehensive feature set for profiling user's behavioral activities in office automation system. The feature set provides comparative assessment between multiple observations in a window period and between multiple users. We apply a range of anomaly metrics to measure the degree of anomaly. User intervention can be added to adjust weights of different anomaly metrics.

However, there exist some problems that need to be explored further. Due to the data missing and the limitation of available data volume, this paper analyzes short-term user behaviors in an offline way. It is a challenge task that use these data to monitor real-time changes of behavioral activities and detect anomaly dynamically. In the future work, we will improve the way of collecting data and explore the impact of different temporal granularities and we will enrich our dataset with more domain data for profiling user

behavioral activities comprehensively, and detecting anomaly more effectively.

Acknowledgment

We would like thank Jiang Wang for his insightful comments and helpful suggestions on this paper. This work was supported by National Natural Science Foundation of China (Grant No. 71571186 and 71471176) and China Postdoctoral Science Foundation (No. 2016M593018)

References

[1] Hong, J., Kim, J., & Cho, J. (2009). *The Trend of the Security Research for the Insider Cyber Threat*. *Security Technology*. Springer Berlin Heidelberg.

[2] Nellikar, S. (2010). Insider threat simulation and performance analysis of insider detection algorithms with role based models. *University of Illinois at Urbana-Champaign*.

[3] Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2017). Automated insider threat detection system using user and role-based profile assessment. *IEEE Systems Journal, 11(2)*, 503-512.

[4] Myers, J., Grimaila, M. R., & Mills, R. F. (2009). Towards insider threat detection using web server logs, 1-4.

[5] Eldardiry, H., Bart, E., Liu, J., Hanley, J., Price, B., & Brdiczka, O. (2013). Multi-domain information fusion for insider threat detection. *IEEE Security and Privacy Workshops*, *42*, 45-51. IEEE Computer Society.

[6] Anderson, G. F., Selby, D. A., & Ramsey, M. (2007). Insider attack and real-time data mining of user behavior. *Ibm Journal of Research & Development, 51(3.4)*, 465-475.

[7] Nguyen, N., Reiher, P., & Kuenning, G. H. (2003). Detecting insider threats by monitoring system call activity. *Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society*, pp. 45-52).

[8] Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2015). Caught in the act of an insider attack: detection and assessment of insider threat. *Proceedings of IEEE International Symposium on Technologies for Homeland Security.*

[9] Park, J. S., & Giordano, J. (2006). Role-based profile analysis for scalable and accurate insider-anomaly detection. *Proceedings of IPCCC 2006 IEEE International Performance, Computing, and Communications Conference, Vol. 2,* (pp. 7-470).

[10] Senator, T. E., Goldberg, H. G., Memory, A., Young, W. T., Rees, B., & Pierce, R., *et al.* (2013). Detecting insider threats in a real corporate database of computer usage activity. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp.1393-1401). ACM.

[11] Magklaras, G. B., & Furnell, S. M. (2001). *Events: Insider Threat Prediction Tool: Evaluating the Probability of IT Misuse*. Elsevier Advanced Technology Publications.

[12] Aggarwal, C. C. (2013). Outlier analysis. 75-99.

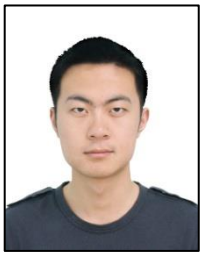
[13] Eberle, W., & Holder, L. (2009). Insider threat detection using graph-based approaches. *Proceedings of Cybersecurity* Applications *& Technology Conference for Homeland Security*, *Vol. 6*, (pp. 237-241). IEEE Computer Society.

[14] Parveen, P., Evans, J., Thuraisingham, B., Hamlen, K. W., & Khan, L. (2012). Insider threat detection using stream mining and graph mining. *Proceedings of IEEE Third International Conference on Privacy, Security, Risk and Trust* (pp. 1102-1110).

[15] Liu, F. T., Ting, K, M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data, 6(1),* 1-39.

[16] Breunig, M. M. (2000). LOF: Identifying density-based local outliers. *ACM SIGMOD Record, 29(2)*, 93-104.

Yilin Wang received the B.S. degree in information management & information systems from Dalian University of Technology, China. She is currently a postgraduate student in the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China. Her current research interest is computer network data mining.



Yun Zhou received his Ph.D. degree in computer science from the Queen Mary, University of London in 2015. He is currently a lecturer in the College of Information Systems and Management at the National University of Defense Technology, Changsha, China. His research focuses on Bayesian methods for prediction, risk management and decision making. He applies these techniques to a wide range of real-world problems, for both academic research and industrial clients. He has published several papers in reputed journals and conferences in this area, including IJAR, UAI, and PGM.

Cheng Zhu received his Ph.D. degree in management science and engineering from the National University of Defense Technology in 2005. He is currently a professor in the College of Information Systems and Management at the National University of Defense Technology, Changsha, China. His research interests include decision support system, machine learning and data mining.