Optimization of Insider Behavior Pattern for Detecting Misuse in the DCD

Sung-Hwan Kim[1], Young-hyun Choi[1], Jung-ho Eom[2], and Tai-Myoung Chung[1]

[1] Internet Management Technology Laboratory,
School of Information and Communication Engineering,
Sungkyunkwan University, Chunchun-dong 300,
Jangan-gu, Suwon, Kyunggi-do, Republic of Korea
{shkim47, yhchoi}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr
[2] Department of Military Studies, Daejeon University,
62 Daehakro, Dong-Gu, Daejeon-si, 300-716, Republic of Korea
eomhun@gmail.com

Abstract. In this study, we researched detection techniques for insider misuse to documents. We proposed detection technique to prevent insider misuse by improved apriori algorithm. Apriori algorithm is an influential algorithm for mining frequent itemsets for association rules. When insider misuse is occurred during paper work in the DCD, it can be detected by apriori algorithm added click ratio. The proposed detection mechanism can reduce the false detection rate by using Apriori algorithm when detecting insider misuse.

Keywords: Insider Misuse, Apriori Algorithm, Insider Threat.

1 Introduction

Insider misuse has become an important issue in enterprise and government. In the past, security techniques for insider misuse were mainly based on monitoring, access control, and encryption. But it is impossible to perfectly protect internal information from insider misuse with access authorization. We proposed the detection techniques for insider misuse to digital document for protecting internal information.

Insider uses his/her legitimate authorization to perform some behavior that is contrary to the security policy. Even if insider has legitimate authorization to access the information or data, he/she uses access authorization to provide the information to someone who does not have authorization or to someone who does deny to access. And insiders use their authorization to extend their privileges in a manner that breaks both the access control and security policies [1].

Our proposed mechanism detects insider misuse by improved apriori algorithm. Apriori algorithm is a method for frequent itemset mining and association rule over the data.

[2] He is a correspondent-author of this paper.

In this paper we will describe related works in section 2 and our proposed algorithm in section 3. We conclude in the last section.

2 Related Works

2.1 Insider Threats

Research report [2] indicated that 58% of insider incidents involved theft or modification of information. Insiders can bypass physical and technical security measures designed to prevent unauthorized access. Most security systems can detect attacks from outside threats, not necessarily internal threats. Insiders know the security policies, procedures, and the associated vulnerabilities. They can access to critical data as using security flaws in the system with no doubt [3].

It is difficult to define insider threat because of lack of a common understanding and definition of the term insider threat. Insiders have a profitable advantage over outsiders who might want to cause damage to an organization. Insiders can bypass physical and logical security safeguards designed to prevent unauthorized access. Insiders are known the security policies and procedures, and technology used in their organizations as well as the vulnerabilities such as inappropriate enforced policies and procedures or technical flaws in IS [4,5].

In recent, research on insider threat and security has been actively conducted. Research focuses on prevention, detection, monitoring, and forensics. In this paper, we focus on an analysis of insider's access pattern for detecting insider misuse.

2.2 Apriori Algorithm

Apriori algorithm is an influential algorithm for mining frequent itemsets for association rules. Apriori algorithm finds frequent itemsets according to a user-defined minimum support. Frequent itemsets can be used to determine association rules which normalize typical patterns in the database [6]. In the first step in the algorithm, it constructs the candidate 1-itemsets. And then it generates the frequent 1-itemsets by deleting some candidate 1-itemsets are lower than the minimum support. Support means the probability of two items at the same time [6,75].

$$Support(A \Rightarrow B) = P(A \cap B) \quad (1)$$

After it finds all the frequent l-itemsets, it joins the frequent 1-itemsets with each other to construct the candidate 2-itemsets. As this step is repeated until candidate itemsets can't be created, we can finally the last frequent itemsets. Support has the disadvantage could not properly measured the association when support value is small. In order to supplement the support's disadvantage, confidence is applied. Confidence means the occurrence probability of B after A.

$$Confidence(A \Rightarrow B) = P(B/A) = P(A \cap B)/P(A) \quad (2)$$

In order to construct all association rules, support is firstly calculated in the itemsets, and then confidence is calculated.

3 The Proposed Algorithm

Our proposed algorithm can optimize insider access pattern using click ratio based Apriori algorithm. If insider does not use the document for a long time, it acts as a negative factor to analyze insider access pattern. It is not easy to determine whether the documents use or not. In this paper, we introduce click ratio for checking whether the documents use or not. Click ratio defines as the frequency of click on the activated document. In other words, it is that how many times click during the activation time. The type of click includes all count by mouse and keyboard. Click ratio is calculated as follows.

$$Click\ Ratio = Click\ Count / Execution\ Time \quad (3)$$

Whenever insider activates the document in the DCD, log server saves all usage history which includes the type of activated document and click count, etc. The calculation method applying click ratio based Apriori algorithm is as follows formula.

$$CA\ Value = (\sum C_i/n) * P(A \cap B) \quad (4)$$

Click ratio plays important role to characterize access pattern to document. If only the activation time apply to the weights, support of a long inactivity time in the document will increase. We applied click ratio to apriori algorithm for complement this disadvantage. Click ratio indicates how often an insider moves the cursor in the activated document. This means that usage percentage of the document is high.

4 Conclusion

We proposed click ratio based apriori algorithm for monitoring insider misuse in the document control domain. Click ratio indicates how often an insider moves the cursor in the activated document. We standardized the movement path from document to document by insider according to the proposed algorithm. We considered the usage time of document by time factor and the insider's activity on document by click ratio. Click ratio is applied as weight factors to apriori algorithm.

5 Acknowledgement

We work was supported by the IT R&D program of MKE/KEIT. [10041244, Smart TV 2.0 Software Platform]

References

1. R. C. Brackney and P. H. Anderson: Understanding the Insider Threat, Proceeding of a March 2004 Workshop, RAND (2004)
2. Dawn Cappelli, Andrew Moore, Randall Trzeckak, and Timothy J. Shimeall: Common Sense Guide to Prevention and Detection of Insider Threats, Carnegie Mellon Software Engineering Institute, (2009)
3. Jonathan White and Brajendra Panda: Automatic Identification of Critical Data Items in a Database to Mitigate the Effects of Malicious Insiders, ICISS 2009, LNCS, Vol. 5905, pp. 208--221 (2009)
4. Mark Maybury, Penny Chase, Brant Cheikes, et al, “Analysis and Detection of Malicious Insiders” International Conference on Intelligence Analysis, (2005)
5. Robert H. Anderson, “Research and Development Initiatives Focused on Preventing, Detecting, and Responding to Insider Misuse of Critical Defense Information Systems”, RAND, (1999)
6. Hyungchang Kang, Kuntak Yang, Chulsoo Kim, Woonjung Rhee, and Bongkyu Lee: A Time-based Apriori Algorithm, Transaction of KIEE, Vol. 59 No. 7, pp. 1327--1331 (2010)
7. Yanbin Ye and Chia-Chu Chiang: A Parallel Apriori Algorithm for Frequent Itemsets Mining, Proceedings of the Fourth International Conference on Software Engineering Research Management and Applications (SERA’06), pp. 87--93 (2006)