An Improved Frequent Pattern Growth Based Approach to Intrusion Detection System Alert Aggregation

Yuxi Sun1*, Xiaomei Chen2

1 School of Computer Science, University of Nottingham, Nottingham, NG7 2RD, United Kingdom
2 Research and Development Department, Wuxi Ke Chuang Zhi Zhen Technology Co., LTD, Wuxi, Jiangsu, 214062, China
*Corresponding author’s e-mail: psyys4@nottingham.ac.uk

Abstract. This paper introduces different approaches to intrusion detection system (IDS) alert aggregation and proposes an improved frequent pattern growth (FP-growth) algorithm for it. This approach can be divided into three parts, which are removal of noisy data, mining association rules and text similarity check. According to the experiment on Snort alarm dataset provided by an enterprise, all the association rules found by the proposed approach are valid. Therefore, compared with FP-growth algorithm, the proposed approach can increase the precision of the result and is useful for alert aggregation.

1. Introduction
In order to check whether malicious behaviour is taking place, either form outside or inside, intrusion detection systems (IDS) have been widely used in enterprises in recent years. By monitoring accesses and data flows in information systems, operators of the information systems can have access to alerts easily and thus, can take action to attacks immediately [1].

However, IDS can generate massive alert data. According to Guo et al. [2], it is possible that several alerts may be generated at the same time by one attack. Most of them are duplicated but there are some cases that multiple different alerts share the same root attack. As a result, the number of alerts produced by IDS is so large that it becomes difficult for administrators to deal with long security reports and find logic link between alerts.

One solution to this problem is alert aggregation [3]. In terms of the definition of alert aggregation, researchers hold different views. Some researchers believe that alerts should be aggregated only if they match in all attributes excluding a little time difference. Other researchers think that all the alerts caused by the same attack should be clustered and aggregated [4]. The approach introduced in this paper is based on the latter concept.

Recently, researchers have made great achievements in developing alert aggregation techniques. In paper written by Ahmed et al. [4], mining-based aggregation is a direction of research. This technique aims to find the relationships or patterns from data sets so that data become easier to understand and analyse. Clustering is one of the commonly used data mining approaches. For instance, Wang and Cui [5] proposed to use genetic algorithms to cluster IDS alerts. In each generation, new population is produced after parent selection, crossover, mutation and applying selection strategy. According to the experiment results, the genetic-algorithm based approach is effective for clustering alerts. However, it may still be hard for
administrators to tell the pattern of alerts. Association rule mining algorithms can also be applied to learn alert pattern. Julisch and Dacier [6] tried applying episode rules to find patterns in alert sequences. Although this technique did find valuable alarm patterns, the cost could be great.

The focus of this paper is on an improved FP-growth based algorithm to IDS alert aggregation. The paper is organized as follows: Section 2 introduces how FP-growth algorithm mines association rules and its drawback in terms of alert aggregation. Section 3 gives a detailed explanation of the components of the improved approach. In section 4, how to do parameter adjustment is explained and analysis of experimental results is presented. Section 5 concludes this paper.

2. FP-growth algorithm description

FP-growth is an Apriori based algorithm, which stores data in Frequent-pattern tree (FP-tree) to mine frequent patterns and association rules. A pattern is defined as frequent if its support count is greater than or equal to a predefined support threshold.

To construct an FP-tree, FP-Growth algorithm requires two passes of the datasets [7]. In the first pass, count the occurrence of each item in each transaction and filter out those items whose support count is less than the minimum threshold. Later, sort transactions remained in the datasets so that items are in the descending order. Then, to generate an FP-tree, scan the current datasets again and add each transaction to the tree one by one. Finally, with the FP-tree, it is possible to mine all the frequent patterns.

Association rules can be achieved provided with frequent patterns and confidence threshold, which represents the relevant degree of items. Based on association rules, we can predict that the occurrence of certain alarms may trigger the occurrence of another alarm. In terms of IDS alert aggregation, such alarms are considered to be associative with each other and should be aggregated.

However, FP-Growth algorithm has one important drawback. From the analysis of the result, the majority of found association rules are irrelevant or redundant [6]. To reduce the number of such invalid association rules, the processes of removing noisy data and checking text similarity are added into the proposed approach.

3. Improved approach description

The improved algorithm consists of 3 stages. The operation sequence is removing noisy data first, then mining association rules and finally checking text similarity.

3.1. Removal of noisy data

According to the analysis of irrelevant association rules, they all include certain type of alerts that have a negative effect on the result. Such alerts have two main features: appear very frequently and have no logic link with other alerts. Considering that FP-Growth algorithm counts the occurrence of alerts to find frequent patterns, the existence of such data can lead to lots of frequent patterns which meet the requirement of support threshold but in fact are irrelevant. Therefore, such type of data is defined as noisy data in our experiment and it is necessary to remove them before mining association rules.

In order to remove noisy data, DBSCAN clustering algorithm is applied. DBSCAN is an unsupervised clustering algorithm based on density. It defines intensity with two parameters, the neighborhood distance threshold and the number of samples within this neighborhood. Since this algorithm can find outliers, it is suitable for removing noisy data. According to the features of noisy data, we cluster alerts in the dataset based on their occurrence. The implementation details are presented in table 1.
Table 1. This table illustrates the implementation of DBSCAN algorithm for removing outliers.

| Input: Initial alarm dataset: $X = \{x_1, x_2, \ldots, x_n\}$; the neighborhood distance threshold: $\text{eps}$; the number of samples within the neighborhood distance: $\text{min\_samples}$ |
|---|---|---|
| Output: The set of noisy data: $R = \{r_1, r_2, \ldots, r_n\}$ |

1: Begin  
2: A list recording the id of each type of alerts and its occurrence: $M = \{M_1\}, M_2\}, \ldots, M_n\}$  
3: For all $x \in X$  
4: For all $m \in M$  
5: If id$(x)$ in $m$  
6: Increment the occurrence of id$(x)$  
7: Break  
8: Else  
9: Add $\{\text{id}(x), 1\}$ into $M$  
10: End  
11: EndFor  
12: For all $m \in M$  
13: If the number of samples in $\text{eps}(m) \geq \text{min\_samples}$  
14: label $m$ as core point  
15: create a new cluster $C$  
16: End  
17: EndFor  
18: For all core points  
19: aggregate multiple clusters into one cluster if possible  
20: EndFor  
21: For all $m \in M$  
22: If $m$ not in any cluster  
23: add $m$ into set $R$  
24: End  
25: EndFor  
26: End

3.2. Mining association rules

After filtering noisy data, FP-Growth algorithm is applied to mine association rules. According to the predefined time interval, multiple alerts are grouped into one transaction. It deserves to be mentioned that duplicate alerts in the same time interval are eliminated. Then, with certain support and confidence threshold, association rules can be obtained according to the process mentioned in section 2.

3.3. Text similarity check

Based on the manual observation of valid association rules, most of the associative alerts share certain similarity on event description. Thus, in order to further increase the precision of association rules, text similarity of alerts description are calculated by cosine similarity equation.
The process has 5 steps. First, split the strings by space. Second, record all different words into a set. Third, encode every word into a digital number. Fourth, construct term frequency vectors $A$ and $B$ for two strings based on the occurrence of each word. Finally, we calculate the similarity of two vectors by using the following cosine similarity equation.

$$\cos \theta = \frac{\sum_{i=1}^{n} (A_i \cdot B_i)}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

After obtaining the similarity degree, we compare it to the predefined text similarity threshold. If it is smaller than the predefined value, the association rule will be eliminated.

4. Experimental analysis

Experiment data uses Snort alarm dataset provided by an enterprise, with a total of 368274 data, including 141 different types of alerts. Each alert is recorded as one line in a json file. It includes the information of the time the alert occurs, the port and ip of the alert, the name, priority, classification, protocol and sid of the event. According to section 3, the proposed approach is made up of 3 steps. This section will explain how to adjust parameters of each component and analyse the final result.

4.1. DBSCAN algorithm

DBSCAN algorithm defines intensity with two parameters, the neighbourhood distance threshold(eps) and the number of samples within this neighbourhood(min_samples). The distance can be measured in different ways, such as Euclidean distance and Manhattan distance. In our experiment, Euclidean distance was adopted and different values of eps and min_samples were tested. The result is illustrated in Figure 1 and Figure 2.

![Figure 1](image1.png)

Figure 1. This figure shows how the value of eps affects finding noisy data.

![Figure 2](image2.png)

Figure 2. This figure shows how the value of min_samples affects finding noisy data.

Figure 1 shows that when the value of the defined neighbourhood distance increases, the number of noisy data decreases. Figure 2 illustrates that when the value of the minimum number of samples within the neighbourhood is greater than a certain number, changes on it do not have obvious effect on clustering.

Based on the professional knowledge of Internet security, there are overall 11 alerts considered as noisy alerts. After comparing the ids of algorithm-found noisy data with the ids of manually labelled noisy data, when eps is equal to 250 and min_samples is equal to 5, the performance of DBSCAN is the best. It finds out 7 noisy data correctly, all of which occur more than 1000 times. However, it also defines 2 noisy alerts, which are not included in the manually labelled noisy data lists. In addition, there are 4 more noisy alerts not being found out by the algorithm, which have a negative effect on the precision of mining association rules.
4.2. FP-Growth algorithm

3 parameters need to be adjusted in FP-Growth algorithm, which are time interval, support and minimum threshold. According to the comparison of experiment results, their effects are as follow.

When the defined time interval value increased, the number of generated association rules grows while the percentage of irrelevant association rules takes up also raises. This is because alarms are more likely to be related to each other when the number of alarms grouped into one transaction becomes bigger. Furthermore, it is observed that when the value of time interval decreases to a certain value, setting it to be an even smaller value will not make any changes on the generation of association rules. In this experiment, this value equals one minute.

When the support threshold is defined to be a greater value, the number of association rules will decrease while there are fewer irrelevant association rules in the final result. However, some associative alarms which occur less frequently will not be mined successfully. This makes sense considering that support threshold describes the least occurrence of alerts. Thus, there is a trade-off between the precision and recall and it is important to choose a proper value for support threshold.

The effect of confidence threshold value on mining association rules is much simpler. As the confidence threshold value grows, the number of association rules will decrease. In this experiment, when time interval equals one minute, support threshold equals 10 and confidence threshold equals 0.6, the proposed approach can find 51 association rules. From the analysis of each rule, it still contains irrelevant association rules because of the influence of noisy data. The precision of the result can be further improved by checking text similarity.

4.3. Text similarity

The process of evaluating text similarity only involves the predefined text similarity threshold value. Based on the observation of experiments, when this value becomes bigger, the number of filtered association rules increases and most of them are irrelevant rules. However, there is also the case that part of the valid association rules are eliminated. In consequence, the value of text similarity threshold can have opposite effects on precision and recall of the final result.

In this experiment, the value of text similarity threshold is set as 0.3. With this step, the overall 51 association rules are reduced to 17 association rules, which are all checked to be valid.

To sum up, all the parameters used in this experiment is presented in table 2.

| Parameter name     | Value |
|--------------------|-------|
| eps                | 300   |
| min_samples        | 5     |
| time interval      | 60s   |
| support threshold  | 10    |
| confidence threshold| 0.6  |
| text similarity threshold | 0.3 |

4.4. Result comparison

Without noisy data removal and text similarity check steps, the number of association rules mined by FP-Growth algorithm is 1414 with the same parameter settings. More than 95 percent of them are irrelevant because of the negative effect caused by noisy data. In contrast, the proposed improved approach only generates 17 association rules but are all valid. For example, one association rule describes the search request and reply of eDonkey, which is a file sharing software.
5. Conclusion
This paper has proposed an improved FP-Growth based algorithm for IDS alert aggregation. First, based on the definition of noisy data, the approach has applied DBSCAN clustering algorithm to filter noisy data. Then, FP-Growth algorithm has been used to mine association rules. In the end, text similarity of related alerts has been checked to reduce the influence of the noisy data, which are not found in the first step. Compared with the FP-Growth algorithm, this approach is able to get rid of most of the noisy data and mine valid association rules. In this way, without professional Internet security knowledge, administrators can easily find unknown logic link between alerts and perform alert aggregation.

Acknowledgments
Thanks to the officers from Wuxi Ke Chuang Zhi Zhen Technology Co., LTD, for their help and support in providing the Snort alarm dataset to conduct the experiment of the proposed approach.

References
[1] Debar, H., Wespi, A. (2001) Aggregation and Correlation of Intrusion-Detection Alerts. In: Wenke, L., Ludovic, M., Andreas, W. (Eds.), Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID '00). Springer-Verlag. London. pp.85-103.
[2] Guo, F., Ye, J., Yu, M. (2009) Design and Implementation of a Distributed IDS Alert Aggregation Model. In: 2009 4th International Conference on Computer Science & Education. Nanning. pp. 975-980.
[3] Mu, C., Shuai, B. (2012) Research on Preprocessing Technique of Alert Aggregation. In: 2012 Fifth International Joint Conference on Computational Sciences and Optimization. Harbin. pp. 597-600.
[4] Ahmed, T., Siraj, M.M., Zainal, A., Din, M.M. (2014) A Taxonomy on Intrusion Alert Aggregation Techniques. In: 2014 International Symposium on Biometric and Security Technologies (ISBAST). Kuala Lumpur. pp. 244-249.
[5] Wang, J., Cui, B. (2009) Clustering IDS alarms with an IGA-based approach. In: 2009 International Conference on Communications, Circuits and Systems. Milpitas. pp.586-590.
[6] Julisch, K., Dacier, M. (2002) Mining Intrusion Detection Alarms for Actionable Knowledge. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton. pp. 366-375.
[7] Shuai, Y., Song, T., Wang, J., Zhan, W. (2018) Hybrid Reliability Parameter Selection Method Based on Text Mining, Frequent Pattern Growth Algorithm and Fuzzy Bayesian Network. Journal of Shanghai Jiaotong University (Science)., 23: 423-428.