An Alert Aggregation Algorithm Based on Iterative Self-Organization

Dapeng Man*, Wu Yang, Wei Wang, Shichang Xuan

Information Security Research Center of Harbin Engineering University, Harbin, 150001, China

Abstract

Considering the problem that intrusion detection systems always produced duplicated alarm information, in this paper we propose an iterative self-organization clustering algorithm. It begins with calculating average value of classes as the new clustering center on the basis of random selection, merging and dividing dynamically, then finish the clustering procedure through the iteration finally. Experimental results with DARPA1999 testing data set show that the clustering method is more excellent than traditional clustering methods in both aggregation rate and error aggregation rate. Besides, it reduces duplicated alarm effectively and provides assistance to further related work.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of Harbin University of Science and Technology Open access under CC BY-NC-ND license.

Keywords: network security, intrusion detection system, alert aggregation, iterative self-organization, clustering algorithm

1. Introduction

Intrusion detection system (IDS) is a new network security technology for protecting computers from attacks. Today it has become a very important part in the network security system. Since the accuracy of IDS is too low and the particle size of alarm clustering is too thin, it produces great amount of useless alarm information which frequently cause small amount of real alerts ignored by administrators. Therefore, it is needed a technology to handle and analyze the massive alarm data effectively and thus direct network security management to develop a reasonable defense strategy.

*Corresponding author. Tel.: +86-451-82589638; fax:+86-451-82589638.
E-mail address: mandapeng@hrbeu.edu.cn
Currently, researches on alert aggregation have made surely achievements. In papers [1][2], they build different attack scenes through training data samples and aggregate alerts based on data mining method. Experiments show that they can surely achieve effective results in specific environment, but it will be difficult to obtain training data under usual network environment and can not adapt the impacts caused by variability of network environment and attacks. Valdes proposed to aggregate alerts by attribute similarity in EMERALD project of U.S. Department of Defense[3][4]. They define similarity membership function to compare the similarity degree of some important properties in alerts, then group and merge the alerts to reduce redundancy ones. ChengPo Mu[5], LinRu Ma[6] also separately make some improvements based on this method. But it has strong dependence on expert knowledge base when selecting associate-attribute, defining membership function, setting attribute weights and associating threshold.

For the purpose of solving flood alarm and duplicated alarm problems of IDS effectively, this paper proposes an Iterative Self-Organizing Data Analysis Techniques Algorithm (ISODATA).

2. Clustering algorithm description

The essence of ISODATA algorithm is to generate an initial class as “seed”, and then iterate clustering automatically according to some discriminant rule. Statistical analysis is used to clustering result of last iteration between two iterations. Select canceling, dividing or merging process according to statistical parameters of existing clusters and continue to the next iteration until it either exceeds the maximum number of iterations or meets the classification parameters. That is the clustering process.

ISODATA is divided into 14 steps. Step 1 to 6: Pre-selecting parameters, initializing clustering of network alarm information and preparing the necessary data for merging and dividing. Step 7: Deciding to merge or to divide in next step. Step 8 to 10: Executing alarm clustering algorithm. Step 11 to 13: Executing alarm merging algorithm. Step 14: Deciding whether to end the algorithm.

• First pre-treat alarm information. Express attribute characteristics as numerical values and normalize them. Therefore, each of alarm information can be seen as a one-dimensional feature vector. Items of any vector represent characteristic properties of alarm information. Consequently, clustering process of alarm information is equivalent to that of one-dimensional feature.

• Randomly select some alarm information as initial values which include clustering centers. In addition some other indicators in initial values should be chose that could be modified artificially in the iteration process. Model samples will be assigned to model classes according to these indicators.

• All alarm information is clustered in accordance with the nearest neighbor rule.

• Calculate clustering function and other indicators in classes. Divide or merge last sets of alarm clustering in accordance with given requirements so as to obtain new clustering centers and at the same time to adjust the number of cluster centers.

• Check whether the alarm clustering result meets requirements. Alert aggregation process is complete if the result is convergence, otherwise go back to step 3.

3. Related concepts

In order to facilitate the understanding of the algorithm, this section reviews some basic concepts defined before and also presents several new ones.

Definition 1: The original alert is defined as the alert that initially is captured by IDS and is not processed. In consideration of a lot of useless information in original alert, we just take parts of main attributes as the characteristic attributes in aggregation algorithm, represented as a 11-tuple Alert (Id, Type, SrcIP, DestIP, SrcPort, DestPort, Time) where id is the unique identity of alarm, Type represents
the type of alarm, \(SrcIP\) and \(DestIP\) represent source IP address and destination IP address, \(SrcPort\) and \(DestPort\) represent source port and destination port and \(Time\) represents alarm generating time.

Definition 2: Super alert is defined as the aggregated alert information. For example, \(\text{Alert.A} (A.Id, A.Type, A.SrcIP, A.DestIP, A.SrcPort, A.DestPort, A.Time)\), \(\text{Alert.B} (B.Id, B.Type, B.SrcIP, B.DestIP, B.SrcPort, B.DestPort, B.Time)\) etc. alarm information are merged into an alert \(\text{Alert.T} (T.Id, T.Type, T.SrcIP, T.DestIP, T.SrcPort, T.DestPort, T.Time)\), then \(\text{Alert.T}\) is called super alert. A super alert represents an attack step or process.

Definition 3: Alert information aggregated by a number of original alerts which have the same attributes and are within the time threshold is called real alert. For example, \(\text{Alert.A} (A.Id, A.Type, A.SrcIP, A.DestIP, A.SrcPort, A.DestPort, A.Time)\), \(\text{Alert.B} (B.Id, B.Type, B.SrcIP, B.DestIP, B.SrcPort, B.DestPort, B.Time)\) etc. alert information are aggregated into an super alert information \(\text{Alert.T}\), If \(\text{Type, SrcIP, DestIP, SrcPort and DestPort}\) of these alert information are the same and \(|B.Time – A.time| < \theta\) (\(\theta\) is the time threshold has been set), then we call \(\text{Alert.T}\) a real alert.

Definition 4: Alert information aggregated by original alerts which have different attributes is called false alert. For example, \(\text{Alert.A} (A.Id, A.Type, A.SrcIP, A.DestIP, A.SrcPort, A.DestPort, A.Time)\), \(\text{Alert.B} (B.Id, B.Type, B.SrcIP, B.DestIP, B.SrcPort, B.DestPort, B.Time)\) etc. alert information are aggregated into an super alert \(\text{Alert.T}\), if at least one attribute of \(\text{Type, SrcIP, DestIP, SrcPort and DestPort}\) of these alert information is not the same but \(|B.Time – A.time| < \theta\) (\(\theta\) is the time threshold has been set), then we call \(\text{Alert.T}\) a false alert.

Definition 5: The number of real alerts contained in a false alert is called false number, represented by \(\text{NumX}\). If a false alert contains more than one real alert, then take the real alert that contains the largest number of original alerts as clustering center to calculate false number.

Definition 6: Aggregation rate is represented as follow: \(Jh = 1 – \frac{\text{NumC}}{\text{NumY}}\). Here \(\text{NumC}\) is the number of original alerts and \(\text{NumY}\) is the number of super alerts.

Definition 7: Error aggregation rate is represented as follow: \(Wj = \frac{\text{NumX}}{\text{NumS}}\). Here \(\text{NumX}\) is false number and \(\text{NumS}\) is the number of real alerts.

4. Experimental analysis

In order to better explain the aggregation performance of ISODATA algorithm, we also perform experiments on membership function aggregation algorithm. Then analyze performance of ISODATA by comparing with membership function aggregation algorithm on the same set of alarm information.

In this section, we use DARPA1999 testing data set for experiment provided by MIT Lincoln laboratory mentioned in [8]. Experiments are performed in Redhat AS5 operating system and Snort 2.8 software platform. The testing data are divided into 5 weeks and 5 days a week. Besides, they are divided into internal and external network data. Here we choose internal network data of fifth week for experimental analysis. The numbers of original alerts are shown in Fig 1.
First, we pretreat alert data in Fig 1. After that we use membership function aggregation method and ISODATA algorithm respectively for data aggregation. Same attribute parameters are set both in the two algorithms including alarm type, source IP, destination IP, source port, destination port and time. The time window of the two algorithms is set 120 seconds according to expertise. Weights of each attribute in membership function aggregation algorithm are shown in table 1.

<table>
<thead>
<tr>
<th>Type</th>
<th>sIP</th>
<th>dIP</th>
<th>sPort</th>
<th>dPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>0.3</td>
<td>0.25</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Initialization parameters of ISODATA are shown in table 2 where $K$ is expectation of clustering number, $\theta_s$ is minimum variance of cluster field, $\theta_c$ is maximum range between clusters and $I$ is iteration times.

<table>
<thead>
<tr>
<th>parameter</th>
<th>$K$</th>
<th>$\theta_s$</th>
<th>$\theta_c$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>2000</td>
<td>0.00001</td>
<td>0.0001</td>
<td>100</td>
</tr>
</tbody>
</table>

ISODATA algorithm iterates clustering according to the position of aggregation centers, therefore time attribute needs to be extracted separately. First aggregate according to previous five attributes, and then divide the final results according to the time window of 120 seconds preconfigured. Finally, we obtain the aggregation results as shown in table 3 and Fig 2.

<table>
<thead>
<tr>
<th>Date</th>
<th>Alert number</th>
<th>Membership</th>
<th>ISODATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon.</td>
<td>196</td>
<td>114</td>
<td>88</td>
</tr>
<tr>
<td>Tue.</td>
<td>2153</td>
<td>182</td>
<td>98</td>
</tr>
<tr>
<td>Wed.</td>
<td>1024</td>
<td>180</td>
<td>93</td>
</tr>
<tr>
<td>Thu.</td>
<td>6902</td>
<td>946</td>
<td>537</td>
</tr>
<tr>
<td>Fri.</td>
<td>2307</td>
<td>206</td>
<td>139</td>
</tr>
</tbody>
</table>

We can obtain the aggregation rate both of membership function algorithm and ISODATA algorithm through aggregation results as shown in Fig 3.

![Fig. 2. Alert aggregation results](image-url)
Consider that internal network data of each day are constant and error aggregation rate is proportional to false number as defined above, consequently we just need to calculate false number of two algorithms. We can obtain ratio of error aggregation rates through the ratio of false numbers of the two algorithms for evaluating the performance of aggregation algorithms. In this paper, we use ratio of ISODATA algorithm and membership function algorithm to measure their aggregation performance as shown in table 4. The error aggregation rate of ISODATA is less than that of membership function algorithm’s if the ratio of their false numbers is between 0 and 1, and vice versa. Obviously in terms of error aggregation rate, ISODATA is more excellent.

Table 4. False number of two algorithms

<table>
<thead>
<tr>
<th></th>
<th>False number</th>
<th>Ratio of EAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Membership</td>
<td>ISODATA</td>
</tr>
<tr>
<td>Mon.</td>
<td>71</td>
<td>40</td>
</tr>
<tr>
<td>Tue.</td>
<td>212</td>
<td>145</td>
</tr>
<tr>
<td>Wed.</td>
<td>115</td>
<td>91</td>
</tr>
<tr>
<td>Thu.</td>
<td>383</td>
<td>281</td>
</tr>
<tr>
<td>Fri.</td>
<td>75</td>
<td>58</td>
</tr>
</tbody>
</table>

Comparisons of the two algorithms through data analysis of Fig 3 and table 4, ISODATA has higher performance both in aggregation rate and error aggregation rate.

5. Conclusion

In this paper, we propose an ISODATA algorithm used in alert aggregation area. Experimental comparisons with the membership function aggregation algorithm show that ISODATA is more excellent in clustering original alerts so as to reduce duplicated alerts effectively.

Also we perceive two weaknesses of ISODATA. First, the convergence speed of ISODATA is too slow. Second, although the subjective factors in ISODATA are less than that in membership function method, it still exists. Consequently the main point of future research is to shorten algorithm’s convergence time. Automatic adjustment of algorithm’s parameters in the learning process through
multiple iterations is also needed so as to optimize the parameters and improve the stability of the algorithm, thus achieve real aggregation with non-artificial intervention.

Acknowledgements

It is a project supported by the Basic Research Foundation of Harbin Engineering University of China (HEUFT09011) and the Fundamental Research Funds for the Central Universities (HEUCF100601).

References


