An IDS Alerts Aggregation Algorithm Based on Rough Set Theory

Ru Zhang¹, Tao Guo¹, Jianyi Liu²,*

¹School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
² School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding author e-mail: Liujy@bupt.edu.cn

Abstract. Within a system in which has been deployed several IDS, a great number of alerts can be triggered by a single security event, making real alerts harder to be found. To deal with redundant alerts, we propose a scheme based on rough set theory. In combination with basic concepts in rough set theory, the importance of attributes in alerts was calculated firstly. With the result of attributes importance, we could compute the similarity of two alerts, which will be compared with a pre-defined threshold to determine whether these two alerts can be aggregated or not. Also, time interval should be taken into consideration. Allowed time interval for different types of alerts is computed individually, since different types of alerts may have different time gap between two alerts. In the end of this paper, we apply proposed scheme on DAPRA98 dataset and the results of experiment show that our scheme can efficiently reduce the redundancy of alerts so that administrators of security system could avoid wasting time on useless alerts.

1. Introduction

Internet is playing a more and more important role in our daily life. Along with its convenience, new threats from Internet have been emerging in endless. To protect devices in Internet from illegal intrusion, IDS and firewalls draw a great attention from researchers.

According to detecting techniques, we can differentiate IDS into two types: signature-based IDS and anomaly-based IDS. Both of signature-based and anomaly-based IDS have their advantage and disadvantage, so under most circumstances different types of IDS should be deployed in different places in order to monitor network system from as many aspects as possible [1]. As a result, a single security events may result in a great number of alerts, especially when it comes to a large-scale network. Administrators may waste time on alerts of no importance and it is harder to reveal real security events overwhelmed by redundant alerts [2]. How to deal with these redundant alerts to make them more accuracy, has become a main topic of security management system. One of the most popular approach is called alerts fusion [3].

Alerts fusion usually consists of three steps:

1) Format multi-source data. To unify different format, several researchers and institutions have come up with several solutions [4], such as CEE [5], CEF [6], IODEF [7].

2) Alerts aggregation. The goal of alerts aggregation is to group alerts that have some attributes in common with each other. We mainly focus on the research of this step.
3) Alerts correlation. The goal of correlation is to reveal the logic connection between alerts and reconstruct the attack scenario. Most researches on this stop focus on multi-step attack correlation, correlation based on causal relationship, and correlation based on attack graph [8].

This paper mainly focuses on alerts aggregation. Previous researches usually lay emphasis on general approaches to aggregate alerts due to some similarities of alerts. However, they ignore that the result of similarities could vary with different attack scenarios. To solve this problem, we apply rough set theory to alerts aggregation and hope to get an ideal aggregation result with calculating feature weight under specific scenarios and taking that time interval between two attacks maybe different under different attack types into account.

This paper will be organized as follow: Section 2 introduces related work. Section 3 applies rough set theory to alerts aggregation and explains the scheme in detail. Then we conduct an experiment with an open dataset and analyze the result of the experiment. Section 5 summarizes the paper and brief future work.

2. Literature Survey
As the core operation of alerts fusion, aggregation has drawn a lot of attention from researchers. Most of them focus on aggregation based on similarities of alerts attributes [9, 10, 11].

H. Debar et al. proposed an abstract framework [12]. First he defined 7 situations and each situation is a subset of one or more attributes from attack source, attack destination or attack type (for example, situation 1 indicates alerts that have same source, destination and attack type, while situation 2 indicates alerts that have same source and destination). Second stage of the scheme is to group alerts to different situations and a single alert may be grouped into more than 1 situation. At last it is up to administrators that which group of aggregation to display. In Debar’s paper, he mainly deals with alerts from 3 aspects: source, destination and attack type. Administrators need to check on different situations for different attack types (for example, for DDOS attacks, we only care about their destination and attack type and usually ignore their attack source, while for R2L attack we need information about source, destination and attack type). Also Debar slices time to 2 seconds as a time window and only within a time window should aggregation be performed.

Maggi F et al. proposed a scheme based on fuzzy set theory [13], in order to deal with time attribute. Maggi claims those alerts that are close in time distance should be aggregated, however it is hard to precisely define what is “close”. Old ways usually cut time into slices then aggregate alerts within these slices. Maggi presents a solution that with the help of fuzzy set theory they can build a mathematical model to describe the time distance, by which a value in time distance can be transformed into an aggregation probability. Then two alerts can be determined to be aggregated or not after comparing the aggregation probability with threshold. Although this approach deals with time attribute dynamically, the authors only present their thoughts rather than go deep into how to build models under different attack scenarios and attack types.

Rough set theory is proposed by Pawlak as a mathematical tool, aiming at dealing with inaccuracy, inconsistent and incomplete information and knowledge [14]. On the basis of Pawlak’s work, Yao et al. proposed decision-theoretic rough set model [15] and Lingras et al. apply decision-theoretic rough set model on clustering [16]. Afterwards, Yu et al. deeply explore the use of decision rough set theory on automatic clustering [17, 18, 19].

3. Proposed Method
3.1. Basic concepts
Definition 1. Tetrad $S = (U, A, V, f)$ describes an information system,

$U$ is a finite, non-empty set call universe, $A$ is a finite, non-empty set called attributes, including condition attributes set $C$ and decision attributes set $D$, $V$ indicates range of attributes, $f: U \times A \rightarrow V$ is an information table, assigning a value from $V$ to each attribute in $A$ and object in $U$.

Definition 2. When classifying the universe $U$, all objects differentiating little from each other should be divided into a same category. This relationship is called equivalence relation. For each attribute sets $P \subseteq A$, equivalence relation can be described as
Equivalence relation is the foundation of rough set theory. All objects with the same equivalence relation cannot be distinguished according to knowledge currently available. We can use equivalence relation to classify the universe $U$:

$$U/\text{IND}(P) = \{[x]_P | x \in U\}$$

where $[x]_P = [x]_{\text{IND}(P)}$ indicates equivalence class classified by relation $P$.

Definition 3. Set $X$ is a non-empty subset on universe $U$, that is $X \subseteq U$. According to knowledge currently available and $I$ as an equivalence relation on universe, the maximizing set consists of objects which can be definitely classified to set $X$, is called $I$–lower approximation of $X$ or positive region, the former is symbolized as $I_*(X)$ while the latter $\text{POS}_I(X)$

$$I_*(X) = \{x \in U : I(x) \subseteq X\}$$

3.2. Overview of the scheme

In our scheme, the set consist of all alerts is defined as the universe $U$. Each alert contains attributes set $A$. Correspondingly, condition attributes set $C$ has 4 attributes (source IP address, source port, destination IP address and destination port), while decision attributes set $D$ has 1 attribute (attack type). Framework of the scheme is shown in Fig.1

Fig.1 Framework of our scheme

The first stage is to acquire alerts. We mainly rely on IDS or firewall to detect attacks and generate alerts.

Preprocessing is to format multi-source alerts with different format. In this paper, we adopt IDMEF [20], a typical format widely-used in transferring information for IDS or security management system.

Next stage is our emphasis. Firstly, we apply concepts of rough set theory to the analysis of alerts aggregation and calculate the importance of alerts’ attributes. Then we normalize the result of attributes’ importance as weight of attributes and compute similarity of two alerts. If their similarity falls within given threshold, then we will go for time interval of the two alerts to determine whether they can be aggregated or not. Each step of aggregation stage will be detailed described in the following paper.

3.3. Weight of attributes

Definition 4. Dependency of decision attributes set $D$ on condition attributes set $C$ is defined as follow:

$$\gamma_C(D) = \frac{\text{POS}_C(U/\text{IND}(D))}{|U|}$$

$\gamma_C(D)$ means, according to information provided by attributes set $C$, how many objects in universe $U$ can be classified to the equivalence class $U/\text{IND}(D)$, in another word, how many alerts in $U$ can be determined to be a specific attack type.

Definition 5. Dependency will change after attribute $C_i$ is removed from condition attributes set $C$, so we can define the importance of attribute $C_i$ as
\[ \sigma(C_i) = \gamma_c(D) - \gamma_{C-C_i}(D) \]

\( \sigma(C_i) \) indicates the influence on classifying an object in universe \( U \) to a specific attack type.

Now we can talk about practical application on alerts aggregation. Alerts set \( U \) classified by decision attributes set \( D \) results in a sequence of subsets \( \{sd_1, sd_2, sd_3 \ldots\} \), each of the subsets contains alerts with a single attack type (for example, we say \( sd_1 \) contains all alerts marked as ipsweep attack while \( sd_2 \) contains pod attack). Alerts set \( U \) classified by condition attributes set \( C \) results in a sequence of subsets \( \{sc_1, sc_2, sc_3 \ldots\} \), and each of the subsets contains alerts that have same condition attributes (for example, \( sc_1 \) contains all alerts launched from 192.168.1.1:11111 to 192.168.1.2:22222, it may include ipsweep attack, pod attack and more). If all alerts in \( sc_1 \) belongs to \( sd_1 \), and that is \( sc_1 \subseteq sd_1 \), we claim that \( sc_1 \) is a lower approximation subset of \( sd_1 \). Sum of all lower approximation subsets of \( sd_1 \) in \( U \), can be called dependency of ipsweep attack on condition attributes set \( C \). With similar steps we can get the result of \( \sigma(C_i) \).

After normalizing the importance of attributes, we can get the weight of attributes.

\[ w_i = \frac{\sigma(C_i)}{\sum_{i=1}^{n} \sigma(C_i)} \]

### 3.4. Similarity of alerts

We compute similarity of two alerts as

\[ \text{similarity} = a_i \times w_i \]

Where

\[ a_i = \begin{cases} 0 & \text{Attribute } C_i \text{ of two alerts is the same} \\ 1 & \text{Attribute } C_i \text{ of two alerts is different} \end{cases} \]

If similarity of two alerts falls within allowed threshold, then we can carry on to the next step.

### 3.5. Time interval

In the format of IDMEF [20], we have two time attributes named start_time and duration. Thus in this paper, we take these two attributes into consideration, as shown in Fig.2.

![Fig.2 Processing of time attributes](image)

In Fig.3-2, \( t_{1s}, t_{2s} \) indicates the start_time of alert 1 and alert 2, while \( t_{1d} \) and \( t_{2d} \) indicates their duration. When \( t_{2s} - (t_{1s} + t_{1d}) \) is less than threshold \( t_{th} \), we can perform aggregation on alert 1 and alert 2.

Considering different attack may come along different threshold, we compute \( t_{th} \) under an attack type (take ipsweep for example) as follows:

\[ \Delta t = \frac{\sum_{i=1}^{n} t_i}{n} \]

\[ t_{th} = \begin{cases} 60 & \Delta t > 60 \\ \Delta t & \Delta t < 60 \end{cases} \]

Where \( t_i \) is the time interval between ith and its next ipsweep alert, and \( n \) is the number of ipsweep alerts.

### 4. Experiment

| Attack category | Attack types | Sip | Sport | Dip | Dport |
|-----------------|--------------|-----|-------|-----|-------|
| DDOS            | neptune      | 0.21| 0.03  | 0.58| 0.18  |
|                 | smurf        | 0.17| 0       | 0.83| 0     |
| Probes          | ipsweep      | 0.65| 0.03  | 0.3 | 0.02  |
|                 | portswipe    | 0.57| 0.13  | 0.26| 0.04  |
| R2L             | dict         | 0.42| 0.02  | 0.35| 0.21  |
|                 | warez        | 0.28| 0.31  | 0.3 | 0.11  |
| U2R             | rootkit      | 0.51| 0.03  | 0.41| 0.06  |
|                 | loadmodule   | 0.49| 0.03  | 0.33| 0.15  |
4.1. Preparation for experiment
We perform experiment on DARPA98 dataset from MIT Lincoln Laboratory. This dataset collects data from 7 weeks, including over 3,000,000 events (1,850,000 alerts among the events) and attack types of the events can be classified into 4 categories and 28 in total number. In this paper, we pick 2 attack types with the largest number in each category and apply our scheme on them.

4.2. Experiment result
According to section 3.3, we first compute weight of each condition attributes.

Our dataset consists of attack types in 4 categories (DDOS attack, probe attack, remote to login attack and user to root attack) and 28 in total number. We compute weight of condition attributes on 2 attack types with the largest number in each category. The result is shown in Table.1.

![Fig.3 Result of alerts aggregation](image)

According to section 3.4 and 3.5, based on weight of attributes, we apply aggregation on these 8 kinds of alerts, and the result is shown in Fig.3.

In addition, we apply scheme in [21] to our dataset and make a comparison on the result of alerts reduction rate, as shown in Table.2

**Table.2 Comparison of alerts reduction rate**

|                  | Reduction rate(0.7 as threshold) | Reduction rate(0.9 as threshold) |
|------------------|----------------------------------|----------------------------------|
| scheme in [21]   | 92%                              | 90%                              |
| our scheme       | 98%                              | 95%                              |

Compared with scheme in [21], we process time interval of two alerts before aggregation while scheme in [21] only slices time as time window. When alerts set contains a large number of objects continuously in time attributes. For example, over 80% of alerts set is Neptune attack and in a period from 15:10 to 16:05 someday, it has over 140, 000 alerts marked as Neptune attack with time interval of 1 second. In our scheme these 140, 000 alerts can be aggregated as 1 but in scheme in [21] we still have more than 500 alerts after aggregation. Apparently our scheme has a higher reduction rate.

4.3. Analysis of experiment result
Result of attributes’ weight almost matches our cognition about network security. DDOS attack has a widely distributed attack source and centrally distributed attack destination, so weight of source IP address is less important than weight of destination IP address. On the contrary, source IP address of probe attack is more important than its destination IP address. As for U2R and R2L attack, which is almost point-to-point attack, weight of source IP address is approximately equal to weight of destination IP address. Usually, attack source port is sort of random while destination port is specific, so weight of destination port is higher. However, this situation is related to the real attack. Smurf...
attack does not have information about port, so weight of source port and destination port is 0. Warez attack contains FTP upload and download and in DARPA98 dataset number of download alerts is outnumbered upload alerts, so its source port is more important.

In the result of aggregation, attack types with a large number of alerts can be efficiently aggregated. In our experiment, DDOS attack has the optimal aggregation result. It is partly because weight of attributes mainly centralized to destination IP address and destination IP address of DDOS attack is not widely distributed. But the mainly reason is that DDOS attack is continuous on time and trigger a lot of alerts in a very short time (such as neptune attack mentioned above).

5. Conclusion
In this paper, we propose a aggregation approach. Based on rough set, we first calculate the weight of attributes and then compute similarity of two alerts with the result of attributes’ weight. At last, aggregation is finally determined by processing time interval of two alerts. Our experiment shows that proposed scheme in this paper can efficiently reduce redundancy alerts.

In the future, we hope to apply our scheme to attacks from real world, in order to widely check the effectiveness of our scheme. Also, we will focus on the next stage of aggregation, hoping to reconstruct the whole attack scenario.

Acknowledgments
This work was supported by the National key Research and Development Program of China (2016YFB0800903), the NSF of China (U1636212, U1636112).

References
[1] SIRAJ A, VAUGHN R B. Alert correlation with abstract incident modeling in a multi-sensor environment[J]. International Journal of Computer Science and Network Security, 2007, 7(8):8-19.
[2] NING P, CUI Y, REEVES D S, et al. Techniques and tools for analyzing intrusion alerts[J]. ACM Transactions on Information and System Security, 2004, 7(2):274-318.
[3] Vauer F, Vigna G, Kruegel C, et al. A Comprehensive Approach to Intrusion Detection Alert Correlation[J]. IEEE Transactions on Dependable & Secure Computing, 2004, 1(3):146-169.
[4] Sapegin A, Jaeger D, Azodi A, et al. Hierarchical object log format for normalisation of security events[C]. International Conference on Information Assurance and Security. IEEE, 2014:25-30.
[5] W Heinbockel, J Judge, R Mcquaid, A Chuvakin, R Marty. Common Event Expression White Paper[OL].June 2008
[6] ArcSight. Common Event Format[OL]. July 2009. http://mita-tac.wikispaces.com/file/view/CEF+White+Paper+071709.pdf.
[7] Danyliw R, Meijer J, Demchenko Y. The Incident Object Description Exchange Format[J]. 2007, 5070.
[8] Hubballi N, Suryanarayanan V. False alarm minimization techniques in signature-based intrusion detection systems: A survey[J]. Computer Communications, 2014, 49(8):1-17.
[9] NING P, CUI Y, REEVES D S, et al. Techniques and tools for analyzing intrusion alerts[J]. ACM Transactions on Information and System Security, 2004, 7(2):274-318.
[10] VALDES A, SKINNER K. Probabilistic alert correlation[A]. Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection[C]. Davis, CA, USA, 2001. 54-68.
[11] JULISCH K. Clustering intrusion detection alarms to support root cause analysis[J]. ACM Transactions on Information and System Security, 2003, 4(6): 443-471.
[12] Wespi A. Aggregation and Correlation of Intrusion-Detection Alerts[C]/ International Symposium on Recent Advances in Intrusion Detection. Springer-Verlag, 2001:85-103.
[13] Maggi F, Matteucci M, Zanero S. Reducing false positives in anomaly detectors through fuzzy alert aggregation[J]. Information Fusion, 2009, 10(4):300-311.
[14] Pawlak Z. Rough Sets International Journal of Information and Computer Sciences[J]. 1982.
[15] Yao YY, Wong S K M, Lingras P. A decision-theoretic rough set model//Ras Z W, Zernakova M, Emrich M Leds. Methodologies for Intelligent System, 5. New York: North-Holland, 1990: 17-24.

[16] Lingras P, Chen M, Miao D. Rough Cluster Quality Index Based on Decision Theory[M]. IEEE Educational Activities Department, 2009.

[17] Yu H, Chu S, Yang D. Autonomous Knowledge-Oriented Clustering Using Decision-Theoretic Rough Set Theory[M]// Rough Set and Knowledge Technology. Springer Berlin Heidelberg, 2010.

[18] Yu H, Liu Z, Wang G. Automatically determining the number of clusters using decision-theoretic rough set[C]// Rough Sets and Knowledge Technology -, International Conference, Rskt 2011, Banff, Canada, October 9-12, 2011. Proceedings. DBLP, 2011:504-513.

[19] Hong Yu, Zhanguo Liu, Guoyin Wang. An automatic method to determine the number of clusters using decision-theoretic rough set[J]. International Journal of Approximate Reasoning, 2014, 55(1):101-115.

[20] Debar H, Curry D, Feinstein B. Rfc 4765: The intrusion detection message exchange format (idmef)[J]. Acta Obstetricia Et Gynecologica Scandinavica, 2007, 90(6):559–563.

[21] Yao Y, Wang Z, Gan C, et al. Multi-source alert data understanding for security semantic discovery based on rough set theory[J]. Neurocomputing, 2016, 208:39-45.