Aggregating Intrusion Detection System Alerts Based on *Row Echelon Form* Concept

Homam El-Taj, Omar Abouabdalla, Ahmed Manasrah,
Moein Mayeh, Mohammed Elhalabi

National Advanced IPv6 Center (NAv6)
UNIVERSITI SAINS MALAYSIA
Penang, Malaysia 11800
(homam, omar, ahmad, moein, elhalabi)@nav6.org

***Abstract*— Intrusion Detection Systems (IDS) are one of the well-known systems used to secure the computer environments, these systems triggers thousands of alerts per day to become a serious issue to the analyst, because they need to analyze the severity of the alerts and other issues such as the IP addresses, ports and so on to get better understanding about the relations between the alerts. This will lead to have a better understanding about the attacks. This paper Investigates the most popular aggregation methods, which deals with IDS alerts. In addition, we propose Time Threshold Aggregation algorithm (TTA) to handle IDS alerts. TTA is based on time as a main component to aggregate the alerts. On the other hand, TTA supports aggregating alerts without threshold, which can be done by setting the threshold value to 0.**

***Keywords*—Intrusion Detection System, False Positive, Redundant Alerts, Alert Aggregation.**

I. INTRODUCTION

The reason behind creating intrusion detection systems (IDS) is because of the huge amount of threats and attacks over the internet and wide networks. In the other, hand IDS triggers huge amount of alerts because of these threats; therefore managing and controlling these alerts need to be studied which led the researchers to investigate these alerts to create methods and techniques such as aggregation to minimize the amount of alerts and group them to make them fewer and to reduce the analyzing process time. Such a progress like this directed to minimize the false positive of IDS too. A good knowledge of IDS and their alerts should be known for better understanding of the aggregation technique.

A. Intrusion Detection System (IDS)

IDS as a system triggers alert or a group of alerts if there is an intrusion of the monitored network based on analyzing the activities, these activities are collected from the network packets stream. IDS has two ways of detecting intrusions either by using *anomaly* [1] technique or *misuse* technique [2] or by merging both techniques starts by checking whether the attack signature saved in the database as a misuse technique then apply the anomaly techniques to check if it is anomaly attack. *Misuse* detecting techniques look for a malicious signature or pattern of the threat based on a set of rules or signatures to detect intrusive behavior while *anomaly* detection technique determines the abnormality of network flow by measuring the distance between the suspicious activities and the norm based on a chosen threshold. The main differences between these two techniques are based on detecting the novel attacks and the false positive rate, where *anomaly* techniques can detect novel attacks and they have a high rate of false positive, *misuse* techniques in the other hand have low rate of false positive without the ability of detecting novel attacks. To differentiate between these two techniques and have a better background you may refer to[3-6].

B. IDS Standard Alerts Format

There is a variety on the sensor types, these sensors trigger a non standard formats of alerts, which led to create the standardization format. One of these standards is the Intrusion Detection Message Exchange Format (IDMEF). This standard was built with Extensible Markup Language (XML) and it has the flexibility to accommodate different needs [7] .

II. AGGREGATION TECHNIQUES

Aggregation technique is one the major parts of IDS studies for grouping and minimizing the alerts to ease the process of analyzing them by removing the redundant alerts. Aggregation techniques group the IDS alerts based on the similarity of the alert features, since some of the alerts related to one event usually they have similar features, so they will be aggregated into one alert. This paper will try to give the answers of the following questions: how to define the alert features? How to calculate the similarity of them?

Valdes [8] proposed an aggregation algorithm by including the five features: source IP addresses, source ports, destination IP addresses, source ports and alert generation time. The compression result of each feature is a value between 0 and 1, while the similarity calculation and the weights of each feature depend on predefined values. But the

This research is sponsored by National Advanced IPv6 Center of Excellence (NAv6)

researcher didn't mention the method of defined the similarity and the weights values. Another proposed solution by [9] was based on the exact matching which gives us the result of 0 or 1, so this algorithm is weak because it reduces a little amount of alerts. Another approach based o [8] algorithm's done by [10] give us slightly different experiment results because he used only the source IP addresses and alert generation time.

Different aggregation technique is introduced by [11], this technique aggregates the alerts by categorizing their features into four classes, then a similarity operator used to compute the similarity of the same features class, but there is no discussion on the computation methods for each features class. Oliver [12] suggested that the alerts should be categorized by attack intensions basis, using the subsequent aggregation processes. Investigating the ideas from the previous proposed methods leads to this proposed algorithm. Time threshold aggregation algorithm (TTA) works on extracting the features' alerts to categories them into groups based on the similarity on these features maintaining the integrity of the alerts' features.

III. PROPOSED ALGORITHM

As mentioned in previous sections, based on some of alerts' features or all of them, the researchers are trying to make their aggregation algorithms without any consideration on the alerts' trigger time. In this proposed algorithm, the merging of any two alerts or more will be based on a threshold value, which should give more accuracy combination results. Figure 3.1 explain the TTA algorithm.

I. *Time Threshold Aggregation algorithm*

TTA works as illustrated in following:

(1) Read IDS alert as n = (R_i, R_2, …, R_n)
(2) Get the first row items as R_i where i = { j_1, j_2,.., j_8 }
(3) Set the Threshold *Th_T* value
(4) Iteration I = n-1
(5) Compare the Rows by $|R_i - R_{i+1}|$
(6) Update the R_{i_8} Value
$R_{i_8} = R_{i+1_8} If\ R_{i+1_1}, R_{i+1_2}, \dots, R_{i+1_8} = 0$
(7) While I ≥ 1 Do
(8) Delete R_{i+1} if(R_{i+1_j} for i_1,…, i_7 = 0 & $|R_{i_8}| \leq Th\_T$)
(9) I = I-1

TTA based on the *Row Echelon Form* [13] Concept, in TTA we conseder the redundunt alerts as false positive alerts if there is an exact matching (R_i = R_{i+1}) for i_1,…, i_8, and if the different is only in the time feature is repeated i_8 we conseder it as a real alert.

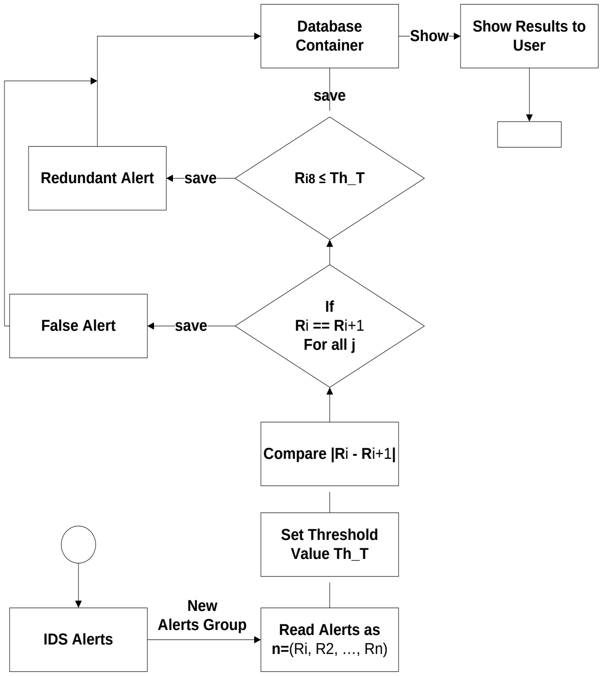

Figure 3.2 TTA Algorithm

To understand the algorithm of ATT, check the following Example:

(1) Let the sample of the A took from the table 3.1 and the Th_T = 2. From table 3.1 we get A_1 = {4, 1, 2, 2, 3, 1, 2, 1}, A_2 = {4, 1, 2, 2, 3, 1, 2, 2}, A_3 = {4, 2, 1, 2, 3, 1, 3, 2}, …, A_{13}{4, 1, 2, 2, 3, 1, 2, 9}.

Table 3.1 Example of A

4	1	2	2	3	1	2	1
4	1	2	2	3	1	2	2
4	2	1	2	3	1	3	2
4	1	2	2	3	1	2	3
7	1	4	2	1	1	1	3
4	1	2	2	3	1	2	3
7	1	4	2	1	1	1	4
4	1	2	2	3	1	2	5
7	3	4	6	1	1	1	6
7	1	4	2	1	1	1	6
4	1	2	2	3	1	2	7
7	1	4	2	1	1	1	8
4	1	2	2	3	1	2	9

(2) We Multiply (A_2, …, A_{13})*-1 then do apply

$$A_{i+1}^{13} = |A_i - A_{i+1}^{13} \qquad (1)$$

A_2 = {-4, -1,- 2, -2, -3, -1, -2, -2}, …, A_{13}{-4, -1,- 2, -2, -3, -1, -2,- 9}. After we apply equation 1 the result will be like this A_1 = {4, 1, 2, 2, 3, 1, 2, 1},

A_2 = {0, 0, 0, 0, 0, 0, 0, 1}, A_3 = {0, 1, 1, 0, 0, 0, 0, 2}, …, A_{13}{0, 0, 0, 0, 0, 0, 0, 3}.

(3) After each time we apply equation 1 we check if A_{i_8} need to be updated by

$$A_{i_8} = A_{i+1_8} \qquad (2)$$

(4) Equation 2 will be used only $A_{i+1_1}, A_{i+1_2}, \dots, A_{i+1_8} = 0$, this equation can be use in the case of A_2, A_4, A_8, A_{11} as {2, 3, 5, 6} in the case of A_{13} A_{13_8} has not been updated because $A_{13_8} > Th\_T$.

(4) We eliminate the zero's rows regarding i_1, …, i_7 to get a new set of A as table 3.2

Table 3.2 New set of A

4	1	2	2	3	1	2	1
4	2	1	2	3	1	3	2
4	2	1	2	3	1	3	2
7	1	4	2	1	1	1	3
7	1	4	2	1	1	1	4
7	3	4	6	1	1	1	6
7	1	4	2	1	1	1	6
7	1	4	2	1	1	1	8
4	1	2	2	3	1	2	9

(5) We repeat step (1, 2, 3, 4, 5) until there is no rows left.

II. *Mathematically Proof :*

If we consider *A* as the alerts' dataset then

$$A = \begin{pmatrix} 4 & 1 & 2 & 2 & 3 & 1 & 2 & 1 \\ 4 & 1 & 2 & 2 & 3 & 1 & 2 & 2 \\ 4 & 2 & 1 & 2 & 3 & 1 & 3 & 2 \\ 4 & 1 & 2 & 2 & 3 & 1 & 2 & 3 \\ 7 & 1 & 4 & 2 & 1 & 1 & 1 & 3 \\ 4 & 1 & 2 & 2 & 3 & 1 & 2 & 3 \\ 7 & 1 & 4 & 2 & 1 & 1 & 1 & 4 \\ 4 & 1 & 2 & 2 & 3 & 1 & 2 & 5 \end{pmatrix} \Rightarrow A = \begin{bmatrix} B \\ \ddot{N} \end{bmatrix}$$

Where *B* is the set of alerts related to the hyper alerts (Produced alerts from *A*) and *N* is the set of alerts representing the false positive alerts (None related).

From the first iteration we got i_1 = [*0 0 0 0 0 0 01*] and from second iteration we got n_1 = [*0 1 -1 0 0 0 1 1*] therefore:

$$\therefore B = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 2 & 2 & 1 & 3 & 2 & 1 \\ 4 & 1 & 2 & 2 & 1 & 3 & 2 & 2 \\ 4 & 1 & 2 & 2 & 1 & 3 & 2 & 3 \\ 4 & 1 & 2 & 2 & 1 & 3 & 2 & 3 \\ 4 & 1 & 2 & 2 & 1 & 3 & 2 & 5 \end{bmatrix}$$

Al

Al
erts
related
l

$$And\ N = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 2 & 3 & 1 & 3 & 2 \\ 7 & 1 & 4 & 2 & 1 & 1 & 1 & 3 \\ 7 & 1 & 4 & 2 & 1 & 1 & 1 & 4 \end{bmatrix}$$

We use the set final set of *N* to the new alerts' dataset, and we keep repeating the algorithm steps until we get empty *N* set $and\ B = [i_1] = [7\ \ 1\ \ 4\ \ 2\ \ 1\ \ 1\ \ 1\ \ 3]$

The final aggregated file *Agg* will be like follow

$$Agg = \begin{bmatrix} 4 & 1 & 2 & 2 & 3 & 1 & 2 & 1 \\ 4 & 2 & 1 & 2 & 3 & 1 & 3 & 2 \\ 7 & 1 & 4 & 2 & 1 & 1 & 1 & 3 \end{bmatrix}$$

Table 4.2 Aggregated Alerts *Agg*

4	1	2	2	3	1	2	1
4	2	1	2	3	1	3	2
7	1	4	2	1	1	1	3

III. *Using Time Threshold Aggregation algorithm on IDS Alerts*

As mentioned in section 1 and 2, alerts contains many features in TTA we focus in 8 features to do the aggregation (*Source IP, destination IP, Source port, destination port, Severity, Protocol, Alert Classification and Time*) so if we took each alert as a row in our algorithm it can give a promising results.

IV. USING TIME AS A MAIN FEATURE

Most of the previous studies didn't take the alert trigger time as one of the extracted features. We believe the time threshold will effects the accuracy of the aggregation result by taking it as one of the aggregation features, based on the alert trigger time, the process of analyzing the alerts will be easier. The analysts would like to know the severity of the alerts based on the amount of the alerts by the same features which this algorithm can show. Based on the threshold Th_T that the user will select; the amount of the aggregated alerts will be changed.

V. Discussion

This paper presented the ATT algorithm for aggregating alerts from any intrusion detection systems. The main advantage of the proposed framework is to improve the alert aggregation process especially when it is related to triggered alert time. The advantages of ATT are: to minimize the amount of alerts, remove the redundant alerts and to remove the false alerts.

Other benefits of the proposed algorithm are: Firstly, to obtain the most benefit from the alerts by making use of the supporting features from the alerts itself which is controlled by the user. Secondly, by analyzing any type of alerts in a standard format, the algorithm provides flexibility to make use of the enriched alerts for aggregating purpose rather than any complicated techniques. Thirdly, this algorithm can ease the study of the alerts severity when they can be related to the aggregated alert groups. Finally, using this algorithm will give us accurate and less number of alerts since it is based on the threshold value which can be modified by increasing or decreasing it. By setting the threshold value to 0 the alerts will be aggregated by exact matching. In other words, the aggregated alerts should be approximately 0 aggregation with the same amount of output alerts, and since it is very hard that two alerts will be triggered in the same time from the same sensor or the same IDS, the amount of the aggregated alerts is high. By increasing the value of the threshold the amount of output alerts will be decreased. After a number of trials the user can tell what are the right value of threshold should be.

VI. Conclusion and Future Work

TTA can be used as an aggregation method to any file containing a group of Items with extracted features. In the stage of programming TTA, it should give the user the ability to choose the number of extracted features from the IDS alerts. TTA can be implemented in using parallel technique for the comparison part between the alerts to give a better time results.

Acknowledgment

This research was supported by the National Advanced IPv6 Center (NAv6) in UNIVERSITI SAINS MALAYSIA (USM).

References

[1] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowledge and Information Systems,* vol. 6, pp. 507-527, 2004.

[2] M. Sheikhan and Z. Jadidi, "Misuse Detection Using Hybrid of Association Rule Mining and Connectionist Modeling," *World Applied Sciences I,* vol. 7, pp. 31-37, 2009.

[3] Y. Liao and V. R. Vemuri, "Use of K-Nearest Neighbor classifier for intrusion detection," *Computers & Security,* vol. 21, pp. 439-448, 2002.

[4] A. Alharby and H. Imai, "IDS false alarm reduction using continuous and discontinuous patterns," Springer, 2005, pp. 192-205.

[5] A. Sundaram, "An introduction to intrusion detection," *Crossroads,* vol. 2, pp. 3-7, 1996.

[6] M. J. Ranum, "False Positives: A User's Guide to Making Sense of IDS Alerts," in *http://searchsecurity.techtarget.com/whitepaperPage/0,293857,sid14_gci903698,00.html*, I. L. IDSC, Ed., 2003.

[7] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)." vol. 2010: March 2007, 2007.

[8] A. Valdes and K. Skinner, "Probabilistic alert correlation," in *the Fourth International Symposium on Recent Advances in Intrusion Detection*, 2001, pp. 54–68.

[9] H. Debar and A. Wespi, "Aggregation and correlation of intrusion-detection alerts," in *4th International Symposium on Recent Advance in Intrusion Detection(RAID) 2001*, 2001, pp. 85-103.

[10] C. Mu, H. Huang, S. Tian, Y. Lin, and Y. Qin, "Intrusion-detection alerts processing based on fuzzy comprehensive evaluation," *Jisuanji Yanjiu yu Fazhan(Computer Research and Development),* vol. 42, pp. 1679-1685, 2005.

[11] F. Autrel and F. Cuppens, "Using an intrusion detection alert similarity operator to aggregate and fuse alerts " in *The 4th Conference on Security and Network Architecture* Batz sur Mer, France, 2005.

[12] O. Dain and R. K. Cunningham, "Fusing a heterogeneous alert stream into scenarios," *Applications of Data Mining and Computer Security,* 2002.

[13] J. Faugère, "A new efficient algorithm for computing Grobner bases (F4)," *Journal of Pure and Applied Algebra.,* vol. 139 pp. 61-88, 1999.