ISSN 2277-3061

# Aggregating IDS Alerts Based on Time Threshold: Testing and Results

Homam Reda El-Taj
Fahad Bin Sultan University, Computer Sconce Department
P.O.Box 15700, Tabuk 71454
Kingdom of Saudi Arabia
homam.eltaj@gmail.com

## ABSTRACT

Every secure system has the possibility to fail. Therefore, extra effort should be taken to protect these systems. Intrusion Detection Systems (IDSs) had been proposed with the aim of providing extra protection to security systems. These systems trigger thousands of alerts per day, which prompt security analysts to verify each alert for relevance and severity based on an aggregation criterion. Several aggregation methods have been proposed to collect these alerts. This paper presents our threshold aggregation system (TAS). Results shows that TAS aggregates IDS alerts accurately based on user demands and threshold value.

## Indexing terms/Keywords

Network security, Intrusion Detection System, Redundant Alerts, Alert Aggregation, Alert Correlation.

## Academic Discipline And Sub-Disciplines

Computer Sconce, Network Security

## SUBJECT  CLASSIFICATION

E.g., Mathematics Subject Classification; Library of Congress Classification

## TYPE (METHOD/APPROACH)

False Positive Reduction by Correlating the Intrusion Detection System Alerts: investigation Study, False Positive Reduction using IDS Alert Correlation Method based on the Apriori Algorithm

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JORNAL OF COMPUTERS & TECHNOLOGY
Vol 11, No.2
editor@cirworld.com
www.cirworld.com, member.cirworld.com

## INTRODUCTION

The use of the Internet and other networks has become essential and extensive. Correspondingly, the threats and intruder activities have become wider and smarter. IDS triggers huge amounts of alerts by detecting these intrusions. Analysts try to analyze these alerts to determine the cause, relations between alerts, severity, and other features of the intrusions. The huge amount of alert data triggered by IDS causes problems during analysis. Several studies have been conducted to help analysts study these alert files with ease.

## RELATED WORK

### Intrusion Detection System (IDS).

The use of the Internet and other networks has become essential and extensive. Correspondingly, the threats and intruder activities have become wider and smarter. IDS triggers huge amounts of alerts by detecting these intrusions. Analysts try to analyze these alerts to determine the cause, relations between alerts, severity, and other features of the intrusions. The huge amount of alert data triggered by IDS causes problems during analysis. Several studies have been conducted to help analysts study these alert files.

Aggregation technique is a major part of correlation techniques. Aggregation has the ability to reduce the complexity involved in alert analysis [1] because the correlation will reduce the amount of alerts after redundancies have been removed through aggregation removal process [2]. The correlation techniques classify the alerts based on features such as IP addresses and port numbers. The higher degree of overall similarity of the alerts in the same class will be correlated [3].

### Correlation Techniques

Ning et al. [1] proposed hyper-alert correlation graphs to discover the attack scenarios based on the specification of individual attacks. Valdes [4] depended on probability approach to correlate the alerts. Debar and Wespi [2] created their correlation method based on duplicates and consequences mechanism. Consequence mechanism deals with a set of alerts linked in a given order within a given time interval. Duplicate mechanism deals with duplicate alerts from different IDS sensors that have the same identical quadruple (source address, source port, target address, and target port).

### Aggregation Techniques

Cuppens [5, 6] developed an aggregation and correlation module (MIRADOR) based on a similarity formulation used to determine the correlation. Yu et al. [7, 8] relied on three functions to aggregate the alerts (alert preprocessing, alert clustering, and collaborative alert merging). The core of this aggregation method is the alert clustering which groups the alerts into different clusters according to the source, target, time, and classification. Zhihong et al. [9] present their aggregation method using the similarity degree function. This function uses probability evaluation before clustering alerts into groups. Liu et al. [10] created an alert aggregation and used it as a sub-model in the NSSA model by adopting an alert aggregation arithmetic. The aggregation model summarizes the alerts base on source IP and attack type after setting a time window. Fan et al. [11] proposed aggregation algorithm based on category and similarity. Fan's algorithm starts with an analysis of attack intentions, followed by aggregation of alerts based on two types of feature sub-groups. Feature sub-groups are divided into categorical such as IPs and ports, and numerical such as packet length. Hofmann and Sick [12] proposed an online alert aggregation based on a probability model. Their work is made up of two phases - online and offline alert aggregations. Offline approach is extended to an on-line approach used for dynamic attack situations.

Valdes proposed a general aggregation algorithm framework by including the five parameters: (Source IP addresses, Source Ports, Destination IP Addresses, Destination Ports, and Alert Generation Time). The compression result of each feature is a value between 0 or 1 result [4]. An enhancement of the Valdes proposal study was conducted by Mu et al, they provided better experiment results because of the use of the source IP addresses and alert generation time only [18].

### Aggregation Based on Time

A new proposal was made by [13] to reduce the false positive in the anomaly detectors by depending on time as one feature to compare with. Their method was based on time intervals. The method suggests that any IDS may recognize an anomaly at time $t_n = t_0 + t_n$ and another anomaly at time $t_h = t_0 + t_h$. Generally, the difference between tn and th will be insignificant. They proposed using the threshold value of $T_{near}$. where if $|t_n - t_h| \leq T_{near}$ then, both attacks belong to each other in terms of SIP, DIP, and IDS. This work was only applied to the anomaly attacks with time intervals; signature-based IDS were not considered.

## THRESHOLD AGGREGATION SYSTEM (TAS)

TAS was built over the threshold aggregation framework TAF [14].

### Alert Standardization

The aggregation methods deal with a heterogeneous alert log file, which receives multiple alert format types from IDSs. The aggregation method must standardize these alerts. Based on TAF, the aggregation will affect only the alerts with minimum quadruple features (SIP, S Port, DIP, and D Port). In the case of UDP, no ports are present. Instead, another set of two features should be present, namely, protocol and the alert type. The best aggregation will work on the eight requirements including the quadruple features shown in Figure 3.1.

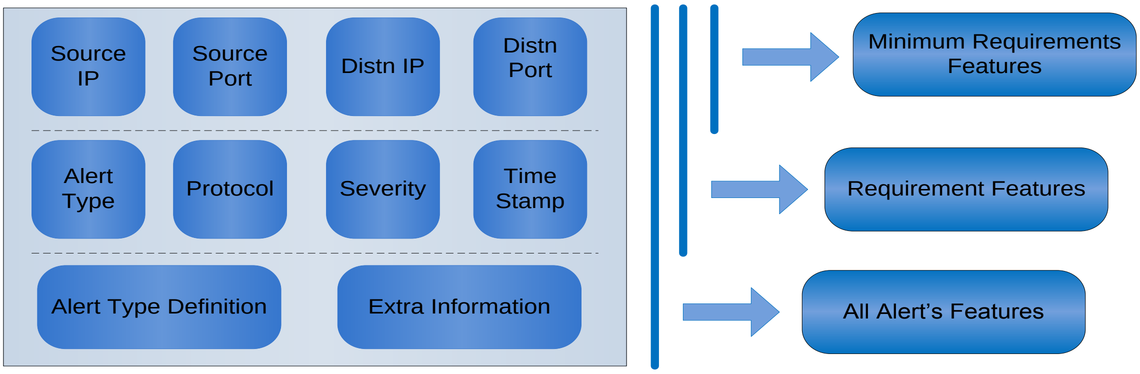

**Figure 3.1 Definition of Alert Requirement Features**

### TAS Method

Alerts got several features. Aggregation will be conducted based on these features, regardless of time difference between the alerts. In TAS, to aggregate two or more alerts, a threshold value should be applied for more accurate combination results, as shown in the next section. Time feature was used so that alerts will not be rejected even with the slightest difference on threshold value. The correlation method built over this aggregation method will have more accurate result alerts without redundant alerts.

Figure 3.2 shows the TAS flowchart. TAS has two types of inputs - the IDS alerts and the user aggregation options. Aggregation will be conducted based on these two inputs. The user will choose the type of aggregation method to be used in IDS alerts. TAS will start processing the IDS alerts by analyzing, manipulating, and then parsing the alerts. A two-phase checking process will be conducted subsequently before deciding whether to drop or save the alert in the database. Finally, the result will be presented or saved in the database depending on user request. TAS got three main components: the data controller, framework core, and aggregation controller. Figure 3.3 shows the architecture of TAS.

### Data Controller

This is the first component in the system responsible for extracting the features from the IDS alerts after receiving them from heterogeneous log file. This data controller has three subcomponents: data analyzer, data manipulator, and data parser.

### Framework Core TASCore

This is the main component in the framework that handles the request from the user and controls the whole framework. After parsing the IDS alerts, the TASCore will start checking the alerts to see whether the features are extracted or not. The alerts are then saved in the database and finally retrieved to the display or the aggregation component (AgC). The TASCore has four subcomponents: alert checker, query generator, database storage controller, and database retrieval.

### Aggregation Controller (AgC)

This component handles the aggregation method that consolidates the alerts into groups based on the chosen subcomponents. These subcomponents will be chosen by the user to control the number of groups that the aggregation method will create. The subcomponents are: severity, protocol, time, alert classification, and alert specification (source and destination IPs and ports). Threshold value is the core of this component. The whole aggregation alert amount will be based on this threshold value.

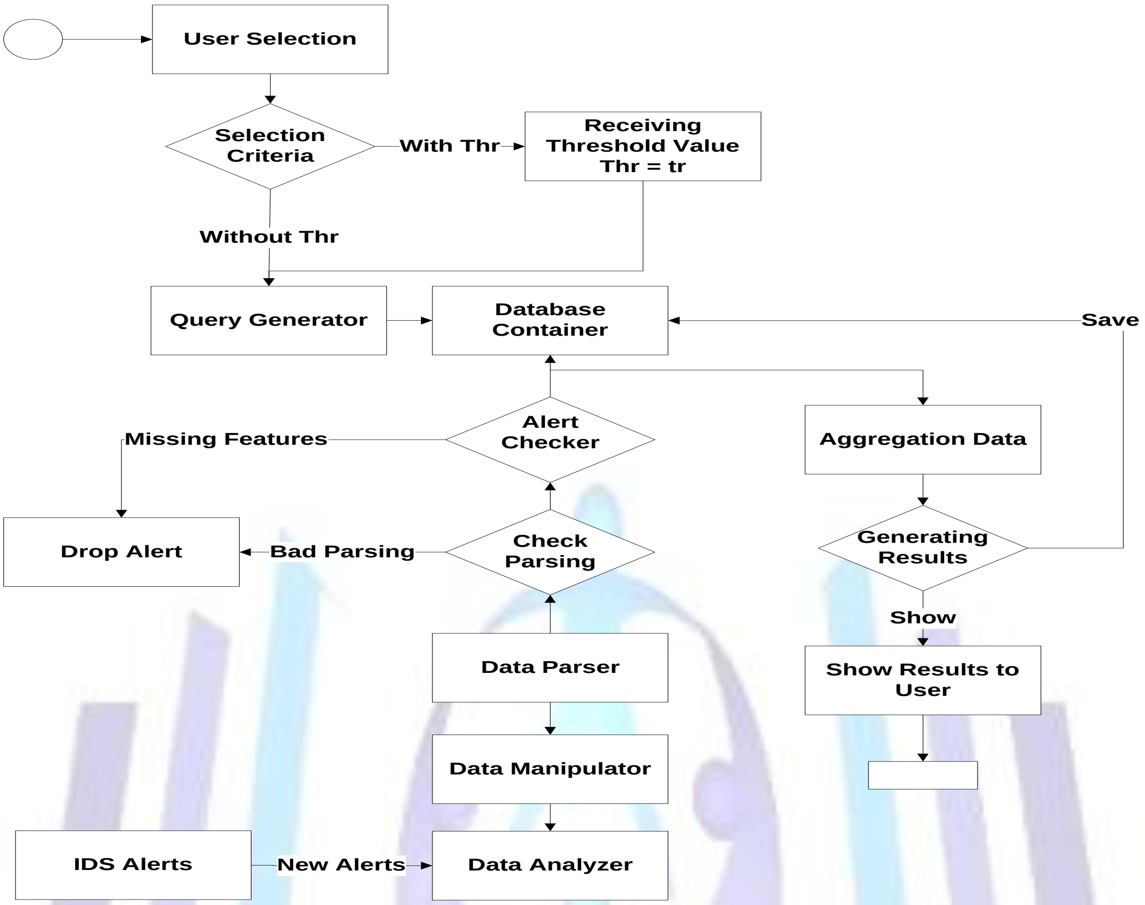

**Figure 3.2 TAS flowchart**

**Table 1. Examples of Alerts**

| | | |
|---|---|---|
| **$\text{Alert}_1$** | **Before** | 12/27-15:31:30.255452 [**] [1:1394:7] SHELLCODE x86 NOOP [**] [Classification: Executable code was detected] [Priority: 1] {UDP} 192.168.1.4:137 -> 192.168.1.1:137 |
| | **After** | 12/27-15:31:30.255452, Executable code was detected, 1, UDP, 192.168.1.4, 137, 192.168.1.1, 137 |
| **$\text{Alert}_2$** | **Before** | 12/27-15:30:44.858790 [**] [1:402:8] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.1.4 |
| | **After** | 12/27-15:30:44.858790, Misc activity, 3 ICMP, 192.168.1.2, 0000 192.168.1.4, 0000 |

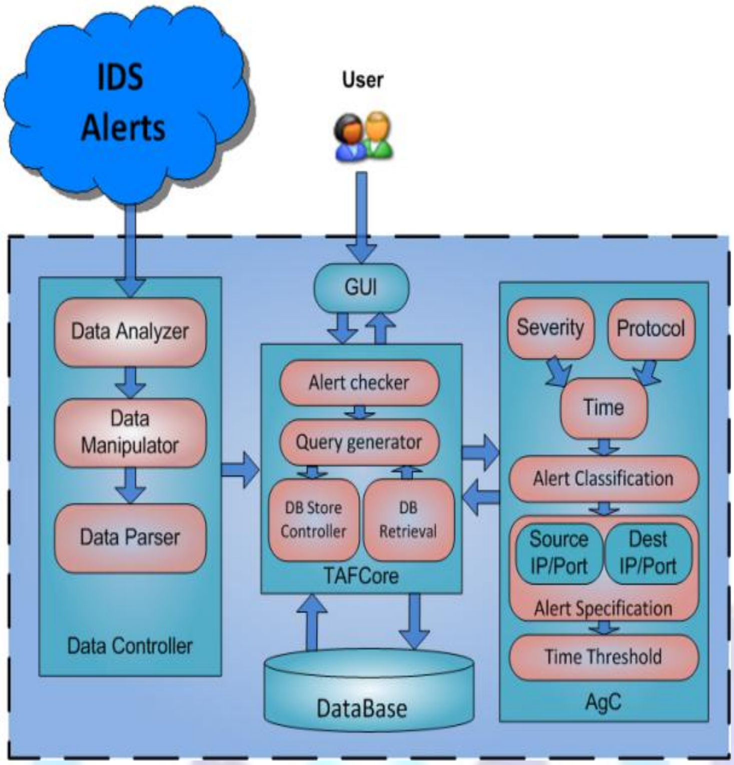

**Figure 3.3 Threshold Aggregation Framework (TAF) [14]**

## IMPLEMENTATION DETAILS

We used Snort in our evaluation [15], which is an open source intrusion detection system that triggers alerts after detecting attacks. The rules of Snort, as well as the process on how Snort works to detect the intrusions, were not given focus because there is no intention to evaluate the performance of Snorts.

The alerts triggered from a multiple IDSs are saved into one heterogeneous file. Each alert in the saved file is represented by several features depending on the version of Snort and how the Snort was configured, the type of IDS sensor, and the alert type (full mode or fast mode) [16].

Focus is given on full mode alerts made up of eight features: date with timestamp, source IP, source port, destination IP, destination port, alert type, severity, and protocol. Table 1 shows two examples of the alerts before and after the 8 features were extracted.

TAS was built using Java programming language and MYSQL database technology. TAS can be used as standalone system to read any IDS alert type if it is contains the minimum requirements. Therefore, TAS can read and analyze heterogeneous log alert files, making the analyst’s job easier. TAS has three main parts: the user which chooses aggregation options and the value of threshold, the IDS alerts to be used in the framework, and the database which holds the processed alerts and the aggregated alerts.

## EXPERIMENTAL RESULTS

In order to evaluate the performance of the alert aggregation, we conducted two types of tests over five different files containing different amounts of alerts (100, 500, 1000, 5000, and 10000). Each file was aggregated four times: with threshold value; without threshold value, with four features, and with eight features. The TAS evaluation is based on the following:

**The result of aggregating alerts without threshold.**

Aggregation without threshold can be conducted in two ways - by selecting all eight features or by selecting some of the features [2, 4, 17]. The quadruple features were chosen, which are the minimum requirements mentioned in Section two. The eight features were then chosen to aggregate the alerts. The study relied on exact matching of the alert features. Table 2 shows the number of alerts in the log file and the number of aggregated alerts in the two processes which are illustrated in Figure 5.1.

**The result of aggregating alerts with threshold.**

Aggregation using threshold can also be achieved in two ways, either by selecting the eight features or by selecting the quadruple features plus the time. Depending on exact matching of the alert features and an experimental threshold value, we performed our testing. Table 3 shows the number of alerts in the log file and the number of aggregated alerts in the two ways which are illustrated in Figure 5.2.

Tables (2 and 3) show that the amounts of incomplete alerts (Inc) are the same since the same data were used for testing. The redundant alerts (Red) which had been deleted by the aggregation is different. Obviously, the amounts of redundant alerts in both cases of 8 features and 4 features are higher than what are presented in Table 3. Depending on how many times the alert is repeated, regardless of the time in Table 2, and by taking the time in Table 3, the aggregation will delete the alert. The amount of aggregated alerts (Agg) in Table 2 for the two cases of 8 features and 4 features are less than what we have presented in Table 3. We depend on the time to aggregate the alert, and thus we should have more accurate alerts to use in the correlation process later.

**Difference between TAS aggregation and other approaches:**

In TAS, we filtered the alerts and chose the correct one that should be used in the second phase of aggregation based on the user demands. The process will decrease the aggregation process time. TAS aggregates the IDS alerts based on user demands and shows full details results based on aggregation alerts, redundant alerts, and incomplete alerts.

**Table 2. The Aggregated Alerts Samples without Threshold**

| **Amount of Alerts** | Selection Method | | | | | |
|---|---|---|---|---|---|---|
| | 8 Features | | | 4 Features | | |
| | Agg | Inc | Red | Agg | Inc | Red |
| **100** | 67 | 7 | 26 | 32 | 7 | 61 |
| 500 | 274 | 30 | 196 | 123 | 30 | 347 |
| 1000 | 679 | 83 | 238 | 289 | 83 | 628 |
| 5000 | 3048 | 214 | 1738 | 1947 | 214 | 2839 |
| 10000 | 6147 | 647 | 3206 | 3417 | 647 | 5936 |

**Table 3. The Aggregated Alerts Samples with Threshold**

| **Amount of Alerts** | Selection Method | | | | | |
|---|---|---|---|---|---|---|
| | 8 Features | | | 4 Features | | |
| | Agg | Inc | Red | Agg | Inc | Red |
| 100 | 81 | 7 | 12 | 73 | 7 | 20 |
| 500 | 423 | 30 | 47 | 362 | 30 | 108 |
| 1000 | 818 | 83 | 99 | 697 | 83 | 220 |
| 5000 | 4310 | 214 | 476 | 3647 | 214 | 1139 |
| 10000 | 8324 | 647 | 1029 | 7314 | 647 | 2039 |

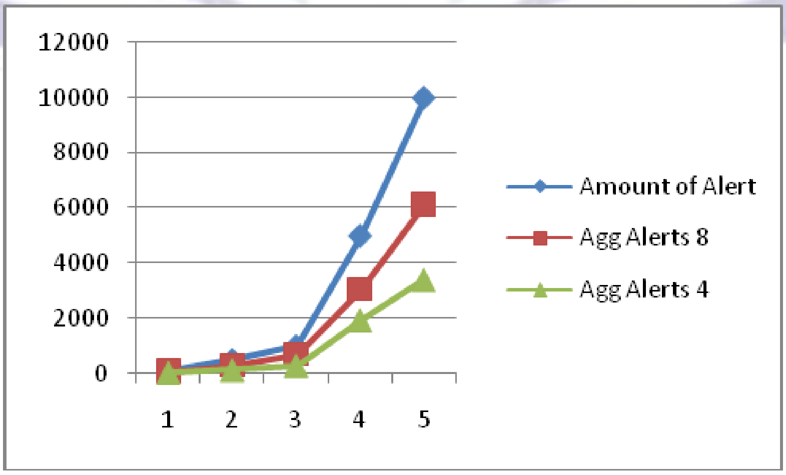

**Figure 5.1 The result of aggregating alerts of 8 and 4 features without threshold** **(Refer to the Appendix for more Details)**

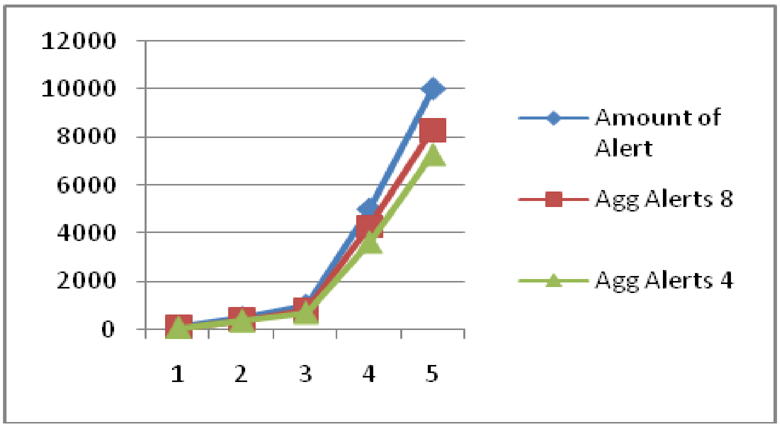

**Figure 5.2 The result of aggregating alerts of 8 and 4 features with threshold value** **(Refer to the Appendix for more Details)**

## CONCLOSION AND FUTURE WORK

Reducing the amount of alerts triggered by IDSs is a challenging area of network security and involves several methods and technologies. In this paper, we focused on aggregating IDS alerts by using the difference in the time stamp between the similar alerts as threshold value. Table 4 and Figure 6.1 show that better results are achieved when IDS alerts are aggregated based on threshold. By using threshold value, the results had been enhanced by 41% with 4 features and 21% with 8 features.

Further research should be conducted to parallelize the TAS to obtain faster results since the heterogeneous alert file sizes are large and take a long time to read. On the other hand, correlation methods can be studied and applied over aggregation method (TAS) to obtain more accurate alerts.

**Table 4. Total amount of aggregated alerts with total reduced alerts average**

| **Amount of Alerts** | **Selection Method** | | | |
|---|---|---|---|---|
| | **Without Th** | | **With Th** | |
| | Avg 8 | Avg 4 | Avg 8 | Avg 4 |
| 100 | 67% | 32% | 81% | 73% |
| 500 | 55% | 25% | 85% | 72% |
| 1000 | 68% | 29% | 82% | 70% |
| 5000 | 61% | 39% | 86% | 73% |
| 10000 | 61% | 34% | 83% | 73% |
| Total Avg | 62% | 32% | 83% | 72% |

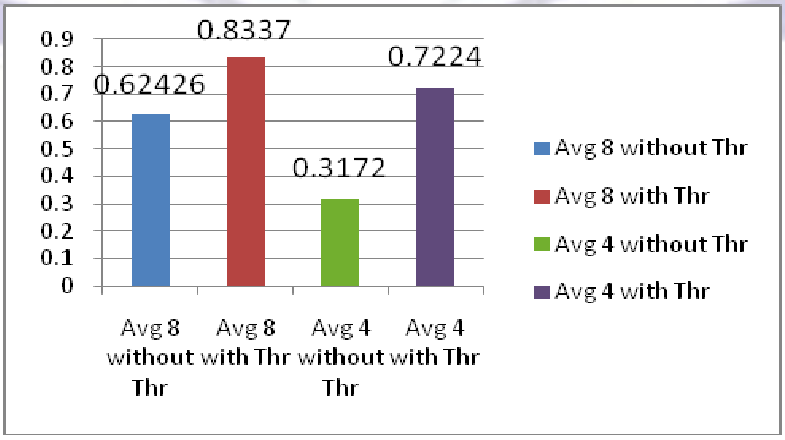

Figure 6.1 The Final result for comparing aggregating with threshold and without threshold

## APPENDIX

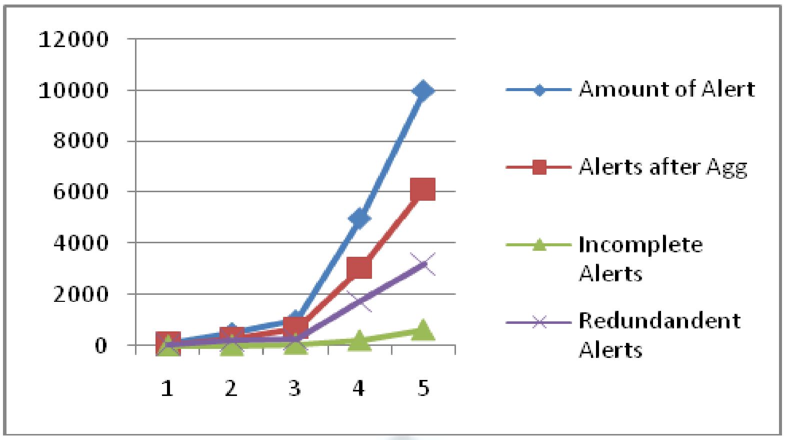

Figure 1 The result of aggregating alerts of 8 features without threshold

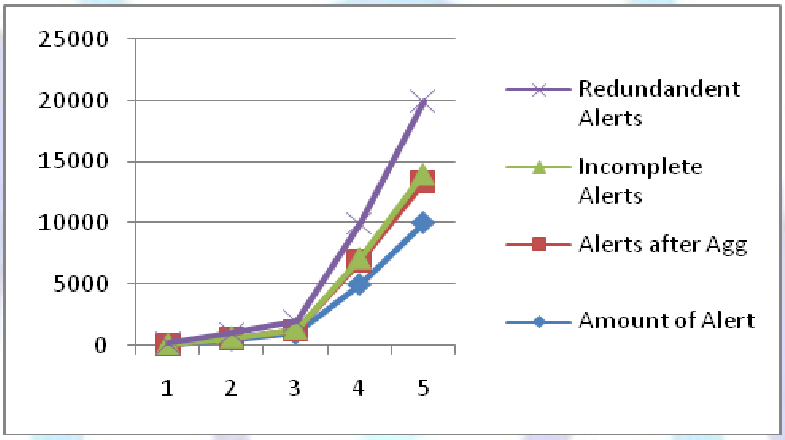

Figure 2 The result of aggregating alerts of 4 features without threshold

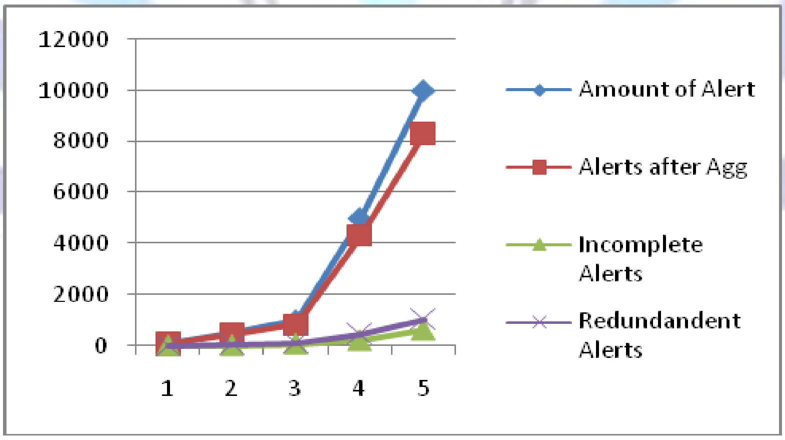

Figure 3 The result of aggregating alerts of 8 features with threshold

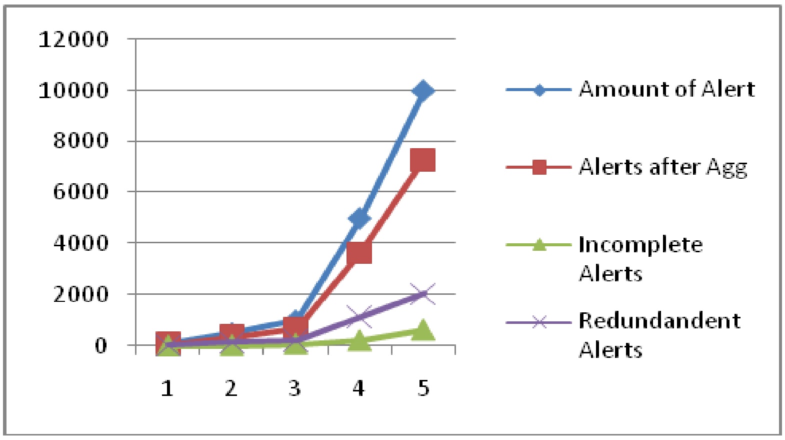

Figure 4 The result of aggregating alerts of 4 features with threshold

## ACKNOWLEDGMENTS

Big thanks to National Advance IPv6 Centre of Excellence (NAv6) for supporting me to finish this work.

## REFERENCES

[1] P. Ning, *et al.*, "Analyzing intensive intrusion alerts via correlation," *Lecture notes in computer science,* pp. 74-94, 2002.

[2] 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.

[3] P. Ning, *et al.*, "Techniques and tools for analyzing intrusion alerts," *ACM Transactions on Information and System Security (TISSEC),* vol. 7, p. 318, 2004.

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

[5] F. Cuppens, "Managing alerts in a multi-intrusion detection environment," 2001.

[6] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," in *IEEE Symposium on Security and Privacy*, Berkeley, California, USA, 2002, pp. 202-215.

[7] J. Yu, *et al.*, "A collaborative architecture for intrusion detection systems with intelligent agents and knowledge-based alert evaluation," in *The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings*, 2004.

[8] J. Yu, *et al.*, "TRINETR: An intrusion detection alert management system," in *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*, 2004.

[9] T. Zhihong, *et al.*, "Alertclu: A Realtime Alert Aggregation and Correlation System," in *International Conference on Cyberworlds 2008*, 2008, pp. 778-781.

[10] X. Liu, *et al.*, "Multiclass Support Vector Machines Theory and Its Data Fusion Application in Network Security Situation Awareness," 2007, pp. 6349-6352.

[11] Y. J. Fan Guo , Yu Min, "Design and Implementation of A Distributed IDS Alert Aggregation Model," in *4th International Conference on Computer Science & Education*, 2009.

[12] A. Hofmann and B. Sick, "On-Line Intrusion Alert Aggregation With Generative Data Stream Modeling," *IEEE Transactions on Dependable and Secure Computing,* 2009.

[13] F. Maggi, *et al.*, "Reducing false positives in anomaly detectors through fuzzy alert aggregation," *Information Fusion,* vol. 10, pp. 300-311, 2009.

[14] Homam El-Taj, *et al.*, "Forthcoming Aggregating Intrusion Detection System Alerts Framework," in *Fourth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2010)*, Venice/Mestre, Italy 2010.

[15] I. Sourcefire. (2010, July). *What is Snort?* Available: http://www.snort.org/

[16] Homam El-Taj, *et al.*, "False Positive Reduction by Correlating the Intrusion Detection System Alerts: investigation Study," *Journal of Communication and Computer,* vol. 7, pp. 25-31, 2010.

[17] [17] C. Mu, *et al.*, "Intrusion-detection alerts processing based on fuzzy comprehensive evaluation," *Jisuanji Yanjiu yu Fazhan (Computer Research and Development),* vol. 42, pp. 1679-1685, 2005.

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

## Author' biography with Photo

Homam Reda El-Taj finished his Bachelor degree in Computer Science (CSCIS) from Philadelphia University Jordan in 2003, then he continued his graduate studies in Universiti Sains Malaysia (USM), he finished his master degree on distributed systems, and PhD on the Network Security.

Homam works as visiting researcher in National Advanced IPv6 Center of Excellence (NAv6) during his work as an assistant professor in Fahad Bin Sultan University (FBSU). Homam was published several articles on Computer Networks field to cover subjects as:

Real time Network Security (Botnets/Worms/viruses), Intelligent Techniques in Detecting Network threats, Advanced Networking Mechanisms and protocols, Intrusion Detection of DOS & DDOS, Intrusion Anomaly Detection Methods, Intrusion Prevention Techniques, Intrusion Prevention, Decision Making, Intrusion Prevention Threats Behavior, Application Network traffic tracing, Network users and misuse detection, Intrusion Detection on QR Code, and Overlay Network.

Currently Homam is supervising some PhD students who are working on fields of Intrusion Detection & Prevention Systems (IDPS) and on the Intrusion Detection on QR Code.