International Journal of Database Theory and Application
Vol.10, No.1 (2017), pp.37-46
http://dx.doi.org/10.14257/ijdta.2017.10.1.04

Multi-source Heterogeneous Data Fusion Method Considering Information Entropy in Large Data Environment

Shujuan Zhang[1] and Zijing Wang[2]

[1]Yunnan College of Business Management, Kunming Yunnan, 650106, China
[2]Department of Scientific and Technical Information, Dah Chong Hong Holdings of Yunnan Region, Kunming Yunnan, 650233, China
E-mail: 27271846@qq.com

Abstract

Massive trivial redundancy alarm information with high error alarm rate, generated by network security defense equipment, causes great difficulty in alarm analysis and understanding. In allusion to the research on this problem, an improved multi-source heterogeneous data fusion scheme is proposed in this paper to comprehensively analyze such attributes as alarm type, source IP, destination IP, destination port and time interval and summarize four rules, thus to dynamically update the time interval threshold value during the fusion process and improve the fusion accuracy. The experiment result shows that such method can efficiently reduce the quantity of the heterogeneous alarm information, and obtain accurate super-alarm data, as well as realize the ability for timely processing the alarm information.

Keywords: *Multi-source heterogeneous; Alarm fusion; Time threshold value; Dynamic update*

1. Introduction

Along with the development of computer technology, there are more and more network security problems. Boundary defense equipment (firewall, IDS, *etc.*), access control, flow monitoring, auditing system, log retention and equipment management service are used together for the security protection of the computer network. However, massive trivial redundancy alarm information with high error alarm rate, generated by network security defense equipment, can cause great difficulty in alarm analysis and understanding, and such alarm information not only directly influences the attack analysis and the timely response to the attack, but also occupies a lot of processing time of the security administrator. Therefore, how to fuse massive heterogeneous alarm information, design an efficient alarm fusion algorithm and establish a uniform alarm platform to really reflect the present network security condition becomes a difficult problem to be solved.

2. Improved Multi-Source Heterogeneous Alarm Fusion Scheme

2.1. Alarm Preprocessing Stage

Due to the non-uniform format, the multi-source heterogeneous alarm data cannot be overall analyzed. IDMEF **Error! Reference source not found.** data model can provide a standard specification for the interoperability among the security equipment. IDMEF format is adopted in this paper to uniformly format these heterogeneous alarm data for future fusion and analysis.

Firstly, it is necessary to analyze various alarm data to describe original alarm A as the following multi-component system form: A (id, detector_id, alarm_class, src_ip, dst_ip. dst_port, start_time, end_time), wherein id represents alarm information ID; detector_id represents alarm generating equipment ID; alarm_class represents the attack type; src_ip represents attack source ip address; dst_ip represents attack target ip address; dst_port represents attack target port; start_time represents attack start time; end_time represents attack end time.

The multi-component system form of super-alarm SA is as follows: SA (id, detector_ids, alarm_calss, src_ip, dst_ip, dst_port, start_time, end_time, alarm_count, orig_alarmids, attack_mode), wherein alarm_count represents the number of the original alarms included in the super-alarm; start_time represents the initial time of an attack; end_time represents the latest end time of an attack; detector_ids represents equipment ID set; orig_alarmids represents the corresponding original alarm ID set which can be used for tracing the original alarms; attack_mode represents the attack mode; A1, A2, A3 and A4 are the four rules introduced subsequently.

The conversion of the multi-source heterogeneous alarm data into above multi-component system form is favorable for uniformly processing the multi-source heterogeneous alarms in order to solve the alarm heterogeneous problem.

2.2. Alarm Fusion

2.2.1. Feature Selection Rule Analysis

According to the analysis of actual network attack patterns **Error! Reference source not found.**, there are following three attack conditions:

(1) Distributed attack from multi-source to single destination, such as denial of service attack;

(2) Attack from single-source to multi-destination, such as various scanning attack activities;

(3) Intrusion activity from single-source to single-destination, such as exploitation of vulnerability and password guess.

According to the attack patterns, the following four rules are defined for the alarm fusion:

- Alert 1: the alarms with the same attribute array (alarm_class, src_ip, dst_ip, dst_port) is regarded as one type, and such description is adopted for a series of alarms probably generated by the same security incident, for example web attack.
- Alert 2: the alarms with the same attribute array (alarm_class, src_ip, dst_ip) is regarded as one type, and such description is adopted for a series of attacks probably from the same source to a certain independent target, for example, port scanning, exploitation of vulnerability, and password guess.
- Alert 3: the alarm fusion with the same attribute array (alarm_class, dst_ip) is regarded as one type, and such description is adopted for the attack behaviors probably from multi-source to a target, for example, denial of service attack (ping Flood, YN Flood, UDP Flood).
- Alert 4: the alarm fusion with the same attribute array (alarm_class, src_ip) is regarded as one type, and such description is adopted for the attack behaviors from the same source to multi-target, for example, scanning attack and Proxy Hunter.

For the alarm fusion methods of the above four attack patterns, the priority is orderly reduced from Alert 1 to Alert 4. The above four rules include the rule preparation for attack type feature and space feature. The time feature will be analyzed in the following paragraph.

2.2.2. Determination of Time Interval Threshold Value and the Dynamic Update Technology Thereof

The original alarm generated by a continuous attack usually appears in a recent period, so the time interval is another important constraint condition of the alarm fusion, and only the alarm data within a certain time interval may be fused.

In most alarm fusion processing, the fixed time interval threshold value **Error! Reference source not found.** and the time interval **Error! Reference source not found.** are commonly adopted as the judgment basis. The key of this method is to determine the maximum time interval t or the time interval for each type of attack. If t or the interval is too small, then "insufficient fusion" will be caused; if t or the interval is too large, then the "excessive fusion" will be caused. Therefore, although it is simple to set the fixed time threshold value at a small overhead, yet the redundancy and inaccuracy problems will be caused and the determination of the threshold value depends on expert knowledge. The relative mean square error of the time interval is introduced in this paper as the dynamic update coefficient to dynamically update the time interval threshold value so as to determine the fusion time interval.

If original alarm sequence is set as *Es* (*e1*, *e2*,…,*en*), the time detected by original alarm *ei* is set as *ts* (*t1*, *t2*,…,*tn*), and $\tau_i = t_{i+1} - t_i$ (i=1, 2, …,n-1) represents the time interval between *ei* and *ei+1*, then the adjacent alarm time interval is $\tau_1, \tau_2, \cdots, \tau_{n-1}$.

The dynamic update formula is as follows:

$$\tau_i = t_{i+1} - t_i \qquad \text{i=1, 2, …,n-1} \tag{1}$$

$$\tau_{avg} = \frac{\sum \tau_i}{n-1} \qquad \text{i=1, 2, …,n-1} \tag{2}$$

$$\sigma(\tau) = \sqrt{\frac{\sum \left(\tau_i - \tau_{avg}\right)^2}{n-1}} \qquad \text{i=1,2,…,n-1} \tag{3}$$

$$\sigma^*(\tau) = \frac{\sigma(\tau)}{\tau_{avg}} \tag{4}$$

$$T = \tau_{avg} + \tau_{avg} \times \sigma^*(\tau) \tag{5}$$

Where τ_{avg} is the mean value of the time interval; $\sigma(\tau)$ is the mean square error of the time interval; $\sigma^*(\tau)$ is the relative square error of the time interval, namely the dynamic update coefficient of the time interval threshold value; T is the time interval threshold value dynamically updated. Dynamic update formula (5) is based on the mean value of the time interval τ_{avg}, and $\sigma^*(\tau)$ is regarded as the coefficient to calculate the final time interval threshold value T.

At the arrival of new alarm *ei*, it is necessary to calculate time interval τ_{i-1} for the former alarm *ei-1* with the same type: if $\tau_{i-1} \le T$ is true, then *ei* and *ei-1* can meet the fusion condition and *ei* should be fused; or else, *ei* should not be fused but should be regarded as a new super-alarm start. For each fusion, the relative mean square error $\sigma^*(\tau)$ of the corresponding time interval will be updated once, so T value will be more and more approximate to the actual time interval threshold value along with the increase of the number of the alarms in original alarm sequence.

The advantages of introducing the method for dynamically updating the time interval threshold value are as follows: 1) this method can will adapt to different attack speeds; 2) this method aims at adjusting the time threshold value according to the time interval fluctuation for automatically updating the threshold value, without depending on the expert knowledge to set the time interval threshold value.

2.3. Alarm Fusion Algorithm

The input of the alarm fusion algorithm is the comprehensive alarm library formed by the preprocessed alarm information generated by the security defense equipment, and the output is the fused super-alarms able to represent massive alarms.

For the first fusion, it is necessary to set a maximum time interval threshold value allowed by users as the default value, or train some data before alarming and preliminarily determine a time interval threshold value as the initial threshold value. During the fusion process, the time interval threshold value is continuously updated. The specific alarm fusion model is as shown in Figure 1.

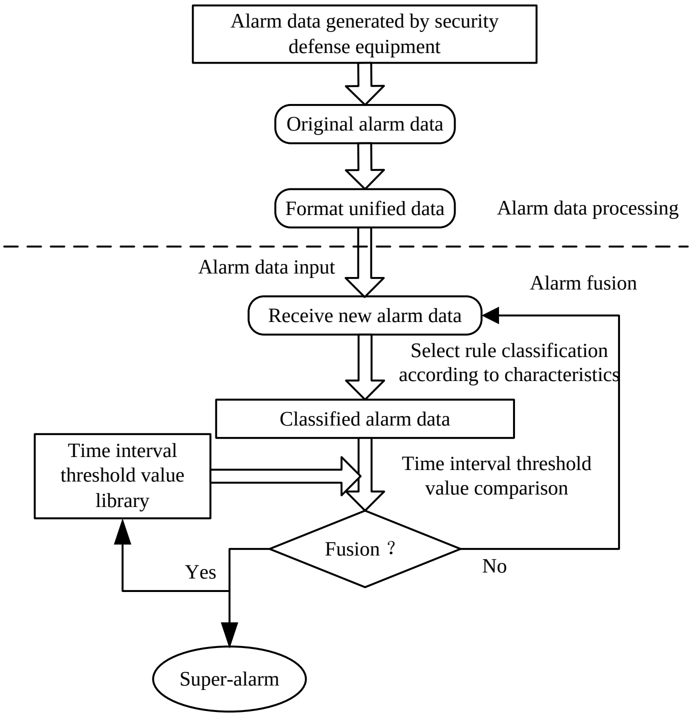

Alarm data generated by security defense equipment
Original alarm data
Format unified data
Alarm data processing
Alarm data input
Alarm fusion
Receive new alarm data
Select rule classification according to characteristics
Classified alarm data
Time interval threshold value library
Time interval threshold value comparison
Fusion ?
Yes
No
Super-alarm

Figure 1. Alarm Fusion Model Figure

In allusion to the calculation for the mean value and the relative square error of the time interval, the optimization algorithm thought in literature [5] is introduced in this paper, and the intermediate alarms are adopted to save the sum and the quadratic sum of the time intervals of the adjacent original alarms in the alarm sequence. At the arrival of a new original alarm, the parameters of the new alarm and the intermediate alarm are directly calculated, and there is no need to calculate the time intervals of all former alarms.

$$TA = \sum \tau_i = \tau_1 + \tau_2 + \cdots + \tau_{n-1} \tag{6}$$

$$TB = \sum \tau_i^{\ 2} = \tau_1^{\ 2} + \tau_2^{\ 2} + \cdots + \tau_{n-1}^{\ 2} \tag{7}$$

$$TA' = \sum \tau_i + \tau_n = TA + \tau_n \tag{8}$$

$$TB' = \sum \tau_i^{\ 2} + \tau_n^{\ 2} = TB + \tau_n^{\ 2} \tag{9}$$

The mean value of the new sequence is as follows:

$$\tau'_{avg} = \frac{TA'}{n} \ ; \ \sigma(\tau) = \sqrt{\frac{1}{n}\left(TB' - \frac{TA'^2}{n}\right)}$$

$$\sigma^*(\tau) = \frac{\sigma(\tau)}{\tau'_{avg}} \tag{10}$$

The updated time interval threshold value is as follows:

$$T = \tau'_{avg} + \tau'_{avg} \times \sigma^*(\tau) \tag{11}$$

The multi-component system form of intermediate alarm S is as follows: (id, detector_id, alarm_class, src_ip, dst_ip, dst_port, start_time, end_time, alarm_count, TA, TB, orig_alarmids, T), wherein TA is the sum of the adjacent time intervals of all original incidents; TB is the quadratic sum of the adjacent time intervals of all original incidents; T is the present time interval threshold value.

The algorithm flow chart of the alarm fusion is as shown in Figure 2.

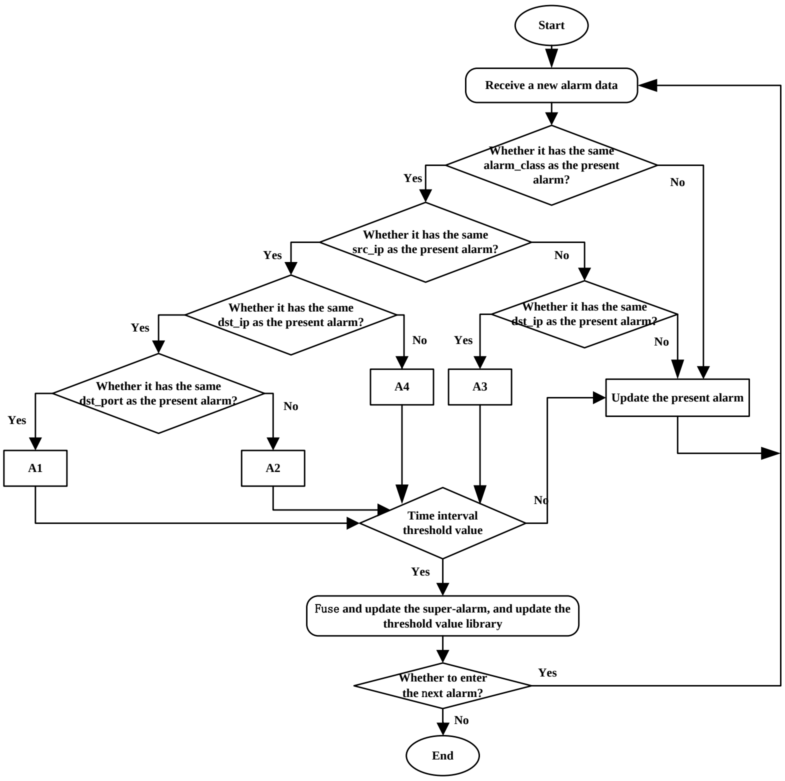

Start
Receive a new alarm data
Whether it has the same alarm_class as the present alarm?
Yes
No
Whether it has the same src_ip as the present alarm?
Yes
No
Whether it has the same dst_ip as the present alarm?
Yes
No
Whether it has the same dst_ip as the present alarm?
Yes
No
Whether it has the same dst_port as the present alarm?
Yes
No
A4
A3
Update the present alarm
A1
A2
No
Time interval threshold value
Yes
Fuse and update the super-alarm, and update the threshold value library
Whether to enter the next alarm?
Yes
No
End

Figure 2. Alarm Fusion Algorithm Flow Chart

The pseudo code of the algorithm is as follows:

```
Public void cluster(){
Match rules A1, A2, A3 and A4;
If Meet the time interval threshold value){
  clusting(currentAlert,nextAlert);// Fuse two alarms
  updateTime（）; // Update time interval threshold value }
else{ currentAlert= nextAlert; // Update present alarm
  continue; //In case of rules satisfaction but time dissatisfaction, interrupt and restart }
}
Public void clusting(Alert a, Alert b){
  Firstly, judge which rule among A1, A2, A3 and A4 does mode belong to
  Update the type of the super-alarm, source ip, destination ip and destination port according to types;
  Update the start time of the super-alarm;
  Update the mode of the super-alarm;
  Increase the super-alarm fusion number;
}
Public void updateTime（）{
  Calculate the time interval;
```

```
        Calculate the present mean value according to intermediate alarms TA and TB;
        Calculate the present relative mean square error coefficient;
        Calculate the present new time interval $T$';
    alarm_A1.T=$T'$; // Update the time threshold value of the corresponding type in the time interval threshold value library;}
```

3. Experiment and Result Analysis

In order to verify the effectiveness of the algorithm, the simulation environment is correspondingly established in the laboratory. Figure 3 is the topological graph of the simulation environment, wherein the virtual machine (Windows 2000 system) established through the virtual platform of the laboratory is adopted as the target machine to provide FTP service, Web service, database service and terminal service; the attack host is installed with two virtual machines of which the systems are respectively Windows 2000 Professional and Backtrack5, for attacking the target machine; the intrusion detection system is installed with Fedora17 and Snort 2.9.4.1 and is configured with corresponding rules; the firewall is also configured with corresponding rules and is installed on the main line; moreover, the log server is configured in the network for collecting syslog alarm logs generated in the network. These tools, Nmap, Metasploit, X-Scan and Hping, are mainly selected to attack the target machine in this test, and the attacks generated thereby include: scanning, exploitation of vulnerability, password guess, web attack and denial of service attack.

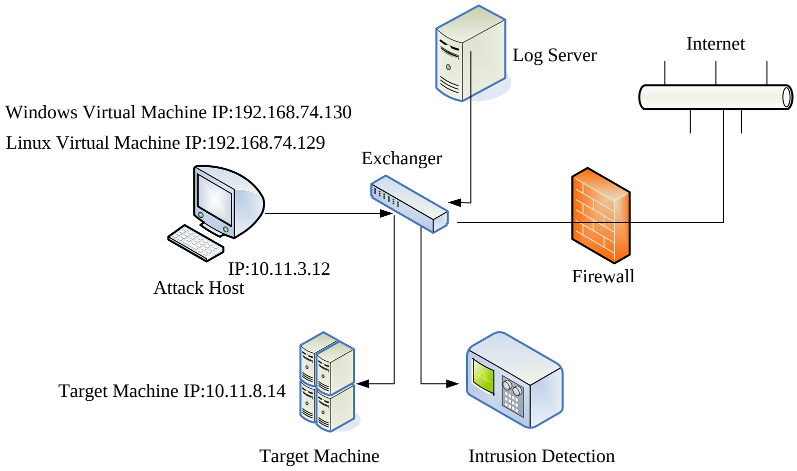
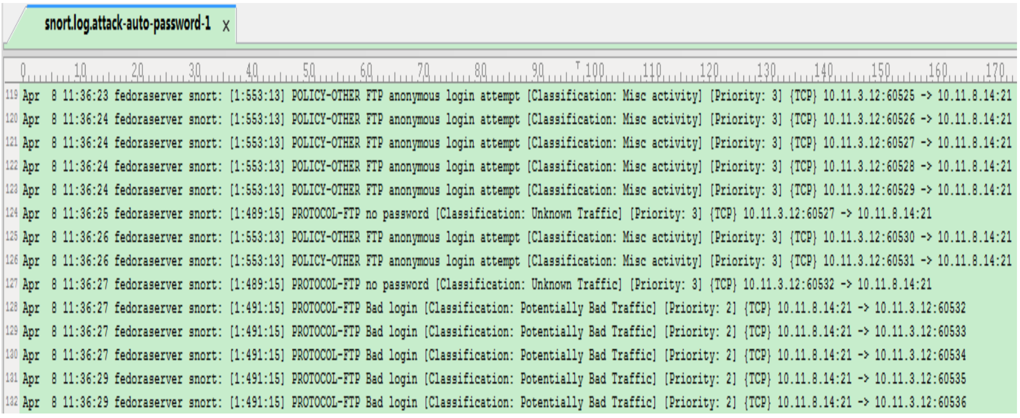

Log Server
Internet
Windows Virtual Machine IP:192.168.74.130
Linux Virtual Machine IP:192.168.74.129
Exchanger
IP:10.11.3.12
Attack Host
Firewall
Target Machine IP:10.11.8.14
Target Machine
Intrusion Detection

Figure 3. Topological Graph of Simulation Environment

In order to measure the fusion degree of the alarm fusion algorithm, we have defined the fusion degree concept, namely:

Fusion Degree = [(Number of Original Alarms – Number of Super-alarms)/Number of Original Alarms] *100%

Figure 4 shows the original alarm data collected by snort for ftp password guess. Firstly, it is necessary to preprocess these alarms to respectively analyze alarm type, equipment ID, source IP, destination IP, destination port, start time and end time; then, it is necessary to implement fusion analysis for the preprocessed data. Table 1 shows the fused super-alarm data. In the super-alarm attributes, not only the original alarm data attributed, but also the number of alarms, the alarm pattern and original alarm id set are reserved in order to trace the original alarm data.



```
119 Apr  8 11:36:23 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60525 -> 10.11.8.14:21
120 Apr  8 11:36:24 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60526 -> 10.11.8.14:21
121 Apr  8 11:36:24 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60527 -> 10.11.8.14:21
122 Apr  8 11:36:24 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60528 -> 10.11.8.14:21
123 Apr  8 11:36:24 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60529 -> 10.11.8.14:21
124 Apr  8 11:36:25 fedoraserver snort: [1:489:15] PROTOCOL-FTP no password [Classification: Unknown Traffic] [Priority: 3] {TCP} 10.11.3.12:60527 -> 10.11.8.14:21
125 Apr  8 11:36:26 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60530 -> 10.11.8.14:21
126 Apr  8 11:36:26 fedoraserver snort: [1:553:13] POLICY-OTHER FTP anonymous login attempt [Classification: Misc activity] [Priority: 3] {TCP} 10.11.3.12:60531 -> 10.11.8.14:21
127 Apr  8 11:36:27 fedoraserver snort: [1:489:15] PROTOCOL-FTP no password [Classification: Unknown Traffic] [Priority: 3] {TCP} 10.11.3.12:60532 -> 10.11.8.14:21
128 Apr  8 11:36:27 fedoraserver snort: [1:491:15] PROTOCOL-FTP Bad login [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.11.8.14:21 -> 10.11.3.12:60532
129 Apr  8 11:36:27 fedoraserver snort: [1:491:15] PROTOCOL-FTP Bad login [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.11.8.14:21 -> 10.11.3.12:60533
130 Apr  8 11:36:27 fedoraserver snort: [1:491:15] PROTOCOL-FTP Bad login [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.11.8.14:21 -> 10.11.3.12:60534
131 Apr  8 11:36:29 fedoraserver snort: [1:491:15] PROTOCOL-FTP Bad login [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.11.8.14:21 -> 10.11.3.12:60535
132 Apr  8 11:36:29 fedoraserver snort: [1:491:15] PROTOCOL-FTP Bad login [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 10.11.8.14:21 -> 10.11.3.12:60536
```

Figure 4. A Segment of Original Alarms Collected by Snort

Table 1. Fused Super-Alarm Data

ID	detector_id	alarm_class	src_ip	dst_ip	dst_port	start_time	end_time	alarm_count	attack_mode	Orig_alarmids
103240	IDS001	POLICY-OTHER FTP anonymous login attempt	10.11.3.12	10.11.8.14	21	11:36:23	11:36:26	7	A1	119;120;121;122;123;125;126
103241	IDS001	PROTOCOL-FTP no password	10.11.3.12	10.11.8.14	21	11:36:25	11:36:27	2	A1	124;127
103242	IDS001	PROTOCOL-FTP Bad login	10.11.3.12	10.11.8.14	21	11:36:27	11:36:29	5	A1	128;129;130;131;132

Through the attack simulation experiment for one week, an optional segment of data is selected as the experimental data. Specifically, the total alarm data volumes of the types with relatively high quantity of alarms during different time periods are as shown in Figure 5.

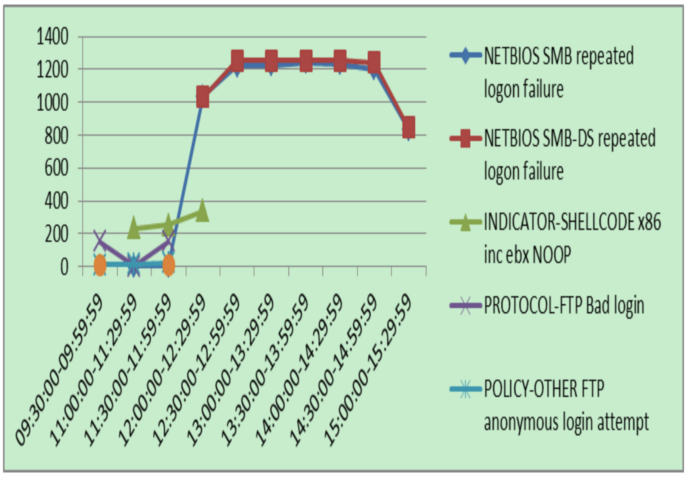


Figure 5. Total Alarm Data Volumes of Several Types in Different Time Periods

The experimental data include 18,232 entries of IDS alarm data, and the proposed fusion method is adopted for the alarm data fusion, and the fusion result is as shown in Figure 6.

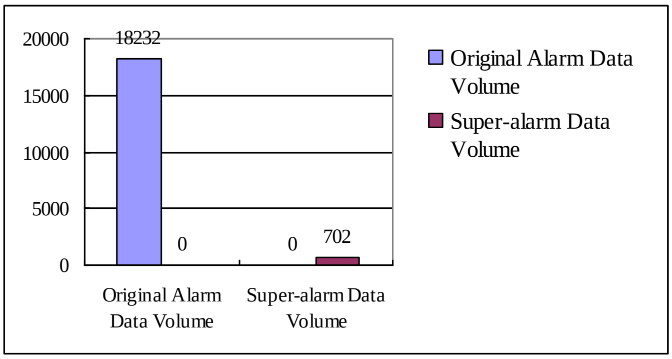


Figure 6. Fusion Result

We have selected FTP related three types of alarms for fusion analysis, and the fusion result is as shown in Figure 7.

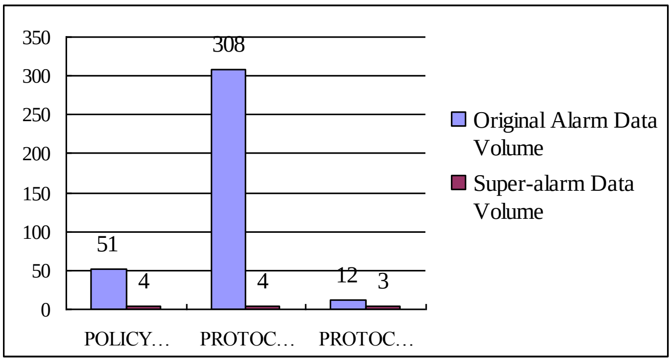


Figure 7. FTP Related Three Types of Alarm Fusion Results

High fusion rate for the fusion of several thousands of NETBIOS related alarms is obtained as shown in Figure 8:

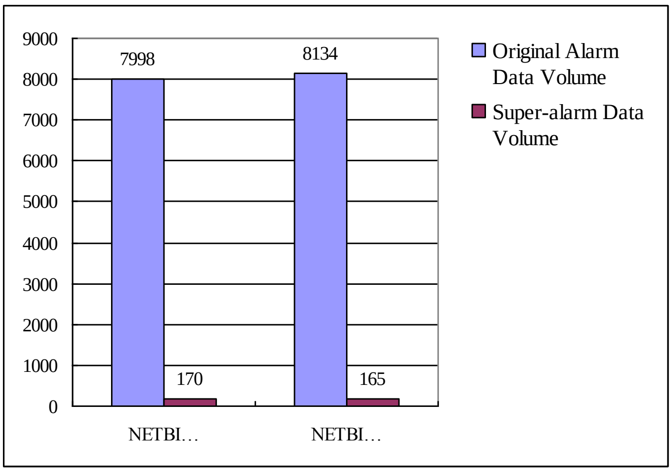


Figure 8. NETBIOS Related Two Types of Alarm Fusion Results

In order to verify the fusion degree of the algorithm in real enterprise network environment, we particularly select the data collected by a certain research institute (note: national secret unit) in one day for the fusion analysis, and the fusion result is as shown in Figure 9:

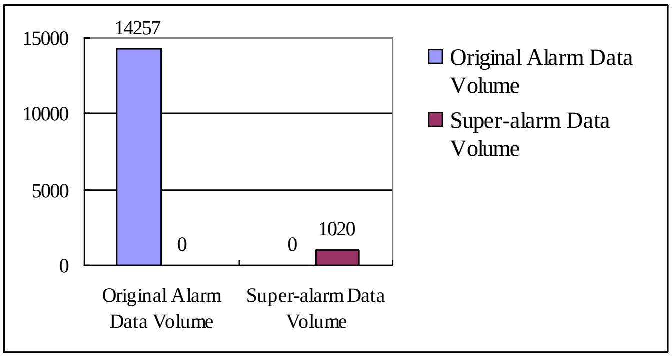


Figure 9. Fusion Results of Alarms Generated by Different Ids of a Research Institute in One Day

The fixed time interval threshold value, the fusion algorithm based on the relative mean square error threshold value of the time interval proposed in literature 5 and the alarm fusion algorithm based on dynamic time interval threshold value proposed in this paper are respectively adopted for the data collected thereby for relevant comparison, and the result is as shown in Table 2.

Table 2. Comparison of Results of Multiple Fusion Methods

Method	Fixed Time Interval Threshold Value Method (Time interval threshold value: 30s)	Method in Literature 5 (relative mean square error threshold value should be given in advance)	The Proposed Dynamic Time Interval Threshold Value Method
Fusion Degree	87.6% (Mean Value)	92.3% (Mean Value)	94.5% (Mean Value)

According to the experiment result comparison shown in Table 2, the proposed method can obtain more accurate super-alarm and higher fusion rate under the same experimental data. The fixed time interval threshold value and the fusion algorithm based on the relative mean square error threshold value proposed in literature [5] need a lot of experiments to determine an optimum threshold value. The time interval threshold value in this paper is dynamically updated, so there is no need to determine a threshold value in advance according to the expert knowledge and the proposed algorithm is a dynamic adjustment method able to meet different network and attack speed requirements, thus more favorable for real-time alarm processing.

4. Conclusion

According to the analysis of the alarm data of such security defense equipment as IDS and firewall, an improved multi-source heterogeneous alarm clustering method is proposed in this paper. Specifically, the data fusion analysis technology is adopted for this method to comprehensively analyze such attributes as alarm type, source IP, destination IP, destination port and time so as to summarize four rules and dynamically update the time interval threshold value during the clustering process. The experiment result shows that the method can obtain accurate super-alarm data and the fusion degree thereof is higher than that of any other method.

References

[1] W. Gu, Z. Lv and M. Hao, "Change detection method for remote sensing images based on an improved Markov random field", Multimedia Tools and Applications, **(2016)**.

[2] Z. Lu, C. Esteve, J. Chirivella and P. Gagliardo, "A Game Based Assistive Tool for Rehabilitation of Dysphonic Patients", 3rd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2015 (VR2015), Arles, France, IEEE, **(2015)**.

[3] Z. Lv, A. Halawani, S. Z. Feng, H. Li and S. U. Rehman, "Multimodal Hand and Foot Gesture Interaction for Handheld Devices", ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 11, 1s, Article 10, **(2014)**.

[4] Y. Lin, J. Yang, Z. Lv, W. Wei and H. Song, "A Self-Assessment Stereo Capture Model Applicable to the Internet of Things", Sensors, **(2015)**.

[5] W. Ou, Z. Lv and Z. Xie, "Spatially Regularized Latent topic Model for Simultaneous object discovery and segmentation", The 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), **(2015)**.

[6] J. Yang, Y. Lin and Z. Gao, "Quality Index for Stereoscopic Images by Separately Evaluating Adding and Subtracting", PloS one, vol. 10, no. 12, **(2015)**, e0145800.

[7] D. Jiang, Z. Xu and Z. Lv, "A multicast delivery approach with minimum energy consumption for wireless multi-hop networks", Telecommunication Systems, **(2015)**, pp. 1-12.

[8] Y. Liu, J. Yang and Q. Meng, "Stereoscopic image quality assessment method based on binocular combination saliency model", Signal Processing, vol. 125, **(2016)**, pp. 237-248.

[9] Z. Lv, A. Tek and F. D. Silva, "Game on, science-how video game technology may help biologists tackle visualization challenges", PloS one, vol. 8, no. 3, **(2013)**, e57990.

[10] X. Zhang, Y. Han, D. Hao and Z. Lv, "ARPPS: Augmented Reality Pipeline Prospect System", 22th International Conference on Neural Information Processing (ICONIP), Istanbul, Turkey. In press, **(2015)**.

[11] Y. Wang, Y. Su and G. Agrawal, "A Novel Approach for Approximate Aggregations Over Arrays", In Proceedings of the 27th international conference on scientific and statistical database management, ACM, **(2015)**.

[12] X. Li, "XEarth: A 3D GIS Platform for managing massive city information", Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015 IEEE International Conference on. IEEE, **(2015)**.

Authors

Shujuan Zhang, holds a MA. Eng. in Software Engineering from University of Electronic Science and Technology of China. Her research interests include Software Engineering, Data Mining. She has published several research papers in scholarly journals in the above research areas. Currently she is an associate professor at the Yunnan College of Business Management.