一种改进的多源异构告警聚合方案*

黄 林[1], 吴志杰[2], 黄晓芳[1], 韦 勇[1], 付智慧[1]
(1. 西南科技大学 计算机科学与技术学院, 四川 绵阳 621010; 2. 中国工程物理研究院, 四川 绵阳 621000)

摘 要: 各类网络安全防御设备产生的大量冗余告警信息非常琐碎、误警率高,给告警的分析和理解造成较大困难。针对这一问题进行研究,提出一种改进的多源异构告警数据的聚合方案,综合分析告警类型、源 IP、目的 IP、目的端口及时间间隔几个属性,总结出四个规则,并在聚合过程中动态更新时间间隔阈值,提高聚合精确度。实验结果表明,这种方法能高效减少异构告警信息的数量,得到精简的超告警数据,并实现了实时处理告警信息的能力。

关键词: 多源异构; 告警聚合; 时间阈值; 动态更新
中图分类号: TP393.08 **文献标志码**: A **文章编号**: 1001-3695(2014)02-0579-04
doi:10.3969/j.issn.1001-3695.2014.02.062

Improved multi-source heterogeneous alert aggregation scheme

HUANG Lin[1], WU Zhi-jie[2], HUANG Xiao-fang[1], WEI Yong[1], FU Zhi-hui[1]
(1. *College of Computer Science & Technology, Southwest University of Science & Technology, Mianyang Sichuan* 621010, *China*; 2. *China Academy of Engineering Physics, Mianyang Sichuan* 621000, *China*)

Abstract: Various types of network security devices generated a large number of redundant alarm information with the high rate of false alarms. Alarm information is very trivial which is more difficult to analysis and understand the alarm. In order to research this problem, this paper proposed an alert aggregation method for multi-source heterogeneous alarm. By analyzing alarm type, source IP, destination IP, destination port and time interval, it summed up four rules, dynamically updated the time interval threshold and improved degree of accuracy. The experimental results show that this method can efficiently reduce the number of heterogeneous alarm information, get simplified super alarm data, and realize the real-time processing ability of the alarm information.

Key words: multi-source heterogeneous; alert aggregation; time threshold; dynamic updates

0 引言

随着计算机技术的发展,网络安全问题日益突出。各类边界防护设备(如防火墙、IDS 等)、访问控制、流量监控、审计系统、日志存留及设备管理服务共同为计算机网络提供安全防护。然而各类安全防御设备产生了大量的告警信息,误警率高,告警信息非常琐碎,对于告警的分析和理解比较困难,这样的告警信息直接影响着对攻击的分析和及时响应,也将占用安全管理员大量的处理时间。所以,如何将大量异构的告警信息进行数据融合处理,设计高效的告警聚合算法,构建统一的告警平台真实反映当前网络安全状况,成为一个亟待解决的难题。

1 相关研究

针对网络入侵告警数据冗余、分析困难的问题,国内外进行了广泛研究,并且已经取得了一些成果。例如,大多数告警聚合利用计算告警属性相似度[1~4]来完成告警数据聚合,但是该类方法需要预先根据专家经验指定相应的权值或系数,为了获得满意的聚合效果,需要反复修改相应的权值或系数,在一定程度上依赖于专家经验。龚俭等人[5]从网络攻击类型、空间和时间属性三个方面对冗余告警的聚合特征进行分析,提出用时间间隔相对均方差模型描述告警事件的时间关联特征,据此给出告警冗余事件消除算法,能实时处理攻击事件并进行冗余消除。这种模型在一定程度上解决了时间间隔变化的问题,然而不同攻击事件序列(即使属于同一种攻击类型)时间间隔的相对均方差未必相同,不同种攻击参数的设定也依赖于专家经验。

本文对文献[5]中的方法作了改进,提出一种改进的多源异构告警聚合方法。将告警聚合特征分为攻击模式约束和时间间隔约束。不同攻击模式具有不同时间间隔约束,并利用文献[5]中的时间间隔相对均方差作为时间间隔动态更新系数,利用时间间隔的平均值作为初始时间间隔阈值,从而使时间间隔阈值自适应网络攻击环境变化。实验结果表明,该方法能有效减少异构告警数量,得到精简的超告警,并实现时间间隔阈值实时动态更新。

收稿日期: 2013-04-27; **修回日期**: 2013-06-05 **基金项目**: 四川省科技支撑计划项目(11ZS2010);四川省教育厅科研资助项目(11ZB108,10ZD1116);西南科技大学博士研究基金资助项目(10ZX7132,11ZX7126)

作者简介:黄林(1988-),女,四川遂宁人,硕士研究生,主要研究方向为网络安全评估、网络安全态势感知技术(314985953@qq.com);吴志杰(1961-),男,河北石家庄人,研究员,主要研究方向为计算机应用;黄晓芳(1976-),女,四川绵阳人,副教授,博士,主要研究方向为密码学、网络安全评估;韦勇(1981-),男,四川成都人,讲师,博士,主要研究方向为网络安全评估、信息内容安全;付智慧(1989-),女,四川资阳人,硕士研究生,主要研究方向为网络安全评估、数字版权管理.

2 改进的多源异构告警聚合方案

2.1 告警预处理阶段

来自多源的异构告警数据在格式上不统一,无法从整体上进行分析,IDMEF[6] 数据模型为安全设备之间的互操作性提供了一种标准规范。本文采用 IDMEF 格式对这些异构的告警数据进行格式化统一,便于以后的聚合和分析。

首先分析各类告警数据,将原始告警 A 描述为如下多元组形式:A(id,detector_id,alarm_class,src_ip,dst_ip,dst_port,start_time,end_time)。其中:id 表示告警信息的 ID;detector_id 表示产生告警的设备 ID;alarm_class 表示攻击类型;src_ip 指攻击源 IP 地址;dst_ip 指攻击目标 IP 地址;dst_port 指攻击目标端口;start_time 表示攻击的开始时间;end_time 表示攻击的结束时间。

超告警 SA 的多元组形式为:SA(id,detector_ids, alarm_class,src_ip,dst_ip,dst_port,start_time,end_time,alarm_count,orig_alarmids,attack_mode)。其中:alarm_count 表示该超告警中所包含的原始告警个数;start_time 是一类攻击的最早开始时间;end_time 是一类攻击的最晚结束时间;detector_ids 为设备 ID 的集合;orig_alarmids 是对应原始告警 ID 的集合,可用于追溯原始告警;attack_mode 表示攻击的模式,即 A1、A2、A3、A4(其中 A1、A2、A3、A4 表示后续介绍的四个规则)。

通过将多源异构的告警数据归约成以上多元组形式,能够很好地对多源异构告警进行统一化处理,从而解决告警异构的问题。

2.2 告警聚合

2.2.1 特征选择规则分析

通过分析实际的网络攻击模式[7],可以分为如下三种攻击情况:

a)多源到单个目的的分布式攻击,如拒绝服务攻击等。

b)单源到多个目的的攻击,如各种扫描攻击活动等。

c)单源到单个目的的入侵活动,如漏洞利用、口令猜测等。

根据攻击的模式,对告警进行聚合定义为如下四条规则:

A1 属性组(alarm_class,src_ip,dst_ip,dst_port)相同的告警聚合为一类,该类描述的可能是同一个安全事件所产生的一系列告警,如 Web 攻击。

A2 属性组(alarm_class,src_ip,dst_ip)相同的告警聚合为一类,该类描述的可能是同一个源对某个单独的目标发起的一系列攻击,如端口扫描、漏洞利用、口令猜测。

A3 属性组(alarm_class,dst_ip)相同的告警聚合为一类,该类可能描述多个源对一个目标的攻击行为,如拒绝服务攻击(ping Flood、SYN Flood、UDP Flood)。

A4 属性组(alarm_class,src_ip)相同的告警聚合为一类,该类描述同一个源对多个目标实施的攻击行为,如扫描攻击、Proxy Hunter。

对于四种攻击模式的告警聚合方式,优先级最高是 A1,其次是 A2、A3,最后是 A4。以上四条规则包含了攻击类型特征和空间特征的规则制定。下面分析时间特征。

2.2.2 时间间隔阈值的确定及其动态更新技术

一次持续性攻击产生的原始告警通常发生在较近的时间段内,因此,时间间隔是告警聚合的另一个重要的约束条件,只有时间间隔在一定范围内的告警数据才可能被聚合。

在大多数告警聚合处理中,采用固定时间间隔阈值[8] 和时间间隔区间[9] 作为判断依据是最常用的方法。使用这种方法的关键问题是为每种攻击类型确定最大时间间隔值 t 或者区间。若 t 值或区间取得过小,会导致欠聚合;若 t 值或区间取得过大,则会导致过聚合。因此,取固定时间阈值虽然简单且开销小,但会产生冗余消除可能不够精确的问题,而且阈值的确定依赖于专家知识。本文引入时间间隔相对均方差作为动态更新系数来动态更新时间间隔阈值的方法以确定聚合的时间间隔。

设原始告警序列 $E_S(e_1,e_2,\cdots,e_n)$,$t_i(i=1,2,\cdots,n-1)$ 为原始告警 e_i 检测到的时间,$\tau_i=t_{i+1}-t_i(i=1,2,\cdots,n-1)$ 表示 e_i 和 e_{i+1} 的时间间隔,则相邻告警时间间隔为 $\tau_1,\tau_2,\cdots,\tau_{n-1}$。

动态更新公式如下:

$$\tau_i=t_{i+1}-t_i \quad i=1,2,\cdots,n-1 \tag{1}$$

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

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

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

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

式中:τ_{avg} 为时间间隔的平均值;$\sigma(\tau)$ 为时间间隔的均方差;$\sigma^*(\tau)$ 为时间间隔的相对均方差,是时间间隔阈值动态更新系数;T 为动态更新的时间间隔阈值。动态更新式(5)以时间间隔平均值 τ_{avg} 作为基础值,以 $\sigma^*(\tau)$ 为系数,从而计算出最终的时间间隔阈值 T。

当新告警 e_i 到达时,计算与前一个同类型的告警 e_{i-1} 的时间间隔 τ_{i-1}。如果 $\tau_{i-1}\leqslant T$,e_i 与 e_{i-1} 满足聚合条件,则将 e_i 聚合;否则 e_i 不进行聚合,作为一个新超告警的起始点。每聚合一次,相应的时间间隔相对均方差 $\sigma^*(\tau)$ 就会更新一次。因此,随着原始告警序列中告警数的增加,T 的值就会越来越趋近于实际的时间间隔阈值。

引入时间间隔阈值动态更新的方法优点在于:a)它能够更好地适应不同的攻击速度;b)这种方法根据时间间隔值的波动程度来调整时间阈值,是一种自动更新阈值的方式,不需要依赖专家知识预先指定时间间隔阈值。

2.3 告警聚合算法

告警聚合算法的输入是各类安全防御设备产生的告警信息经过预处理之后形成的综合告警库,输出是经过聚合之后产生的能代表大量告警的超级告警。

在首次进行聚合时,默认设置一个用户允许的最大时间间隔阈值,或者在进行告警之前,对一部分数据进行训练,初步确定一个时间间隔阈值作为初始阈值。在聚合的过程中,不断更新时间间隔阈值。具体告警聚合模型如图 1 所示。

针对时间间隔平均值与时间间隔相对均方差的计算,引用文献[5]中的优化算法思想,采用中间告警保存来告警序列中相邻原始告警的时间间隔之和、时间间隔平方和。当一个新的原始告警到来时,新告警与中间告警中的参数直接进行计算,无须再计算之前所有的告警时间间隔值。

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

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

$$TA' = \sum \tau_i + \tau_n = TA + \tau_n \quad (8)$$

$$TB' = \sum \tau_i^2 + \tau_n^2 = TB + \tau_n^2 \quad (9)$$

则新序列的平均值为

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

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

更新的时间间隔阈值为

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

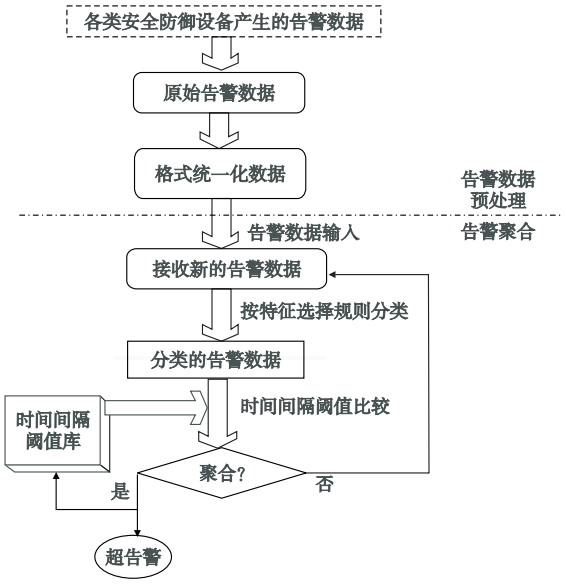

图1　告警聚合模型

中间告警 S 的多元组表示形式为:S(id,detector_id,alarm_class,src_ip,dst_ip,dst_port,start_time,end_time,alarm_count,TA,TB,orig_alarmids,T)。其中:TA 为所有原始事件相邻时间间隔和;TB 为所有原始事件相邻时间间隔平方和;T 为当前的时间间隔阈值。

告警聚合的算法流程如图 2 所示。

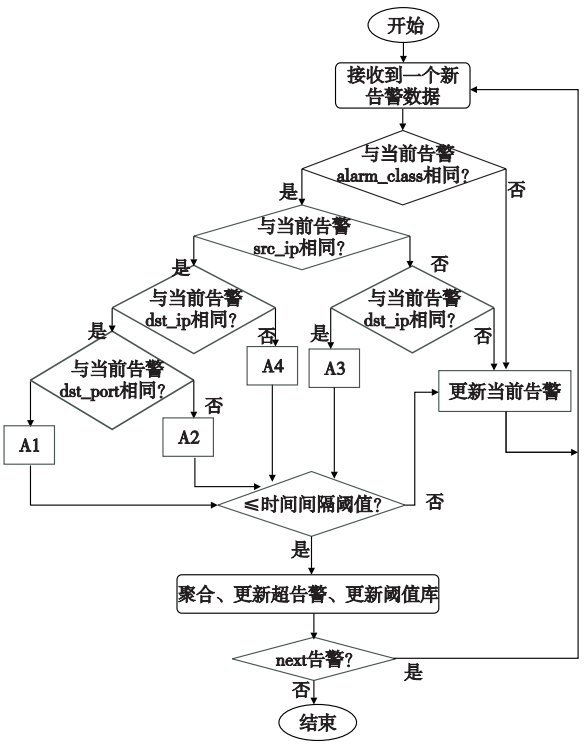

图2　告警聚合算法流程

算法伪代码如下:

```
Public void cluster(){
匹配 A1、A2、A3、A4 规则;
if(满足时间间隔阈值){
  clusting(currentAlert,nextAlert);//聚合两条告警
  updateTime(); //更新时间间隔阈值 }
else{ currentAlert = nextAlert;//将当前告警更新
  continue;
//如果规则满足,但是时间不满足,那么中断,重新开始;}
}
Public void clusting(Alert a, Alert b){
首先判断 mode 是 A1、A2、A3、A4 中哪一个规则;
按类更新超告警的类型,源 IP、目的 IP、目的端口;
将超告警的起始时间更新;
更新超告警的模式;
增加超告警聚合数目;
}
Public void updateTime(){
计算时间间隔;
根据中间告警的 TA、TB 值计算当前平均值;
计算当前相对均方差系数;
计算当前新的间隔时间 T';
alarm_A1.T = T';
//将时间间隔阈值库中对应类型的时间阈值更新;}
```

3　实验及结果分析

为了验证算法的有效性,本文在实验室搭建了模拟环境。图 3 为模拟环境的拓扑图。其中,靶机采用实验室虚拟平台搭建的虚拟机(Windows 2000 系统),提供 FTP、Web 和数据库和终端服务;攻击主机安装两个虚拟机,系统分别为 Windows 2000 Professional 和 Backtrack5,用于对靶机发动攻击;入侵检测系统安装 Fedora17 和 Snort 2.9.4.1,并配置相应规则集;防火墙配置相应规则,安装在干路上;网络中还有搭建日志服务器,用于收集网络中产生的 syslog 告警日志。针对本项目测试,主要选取 Nmap、Metasploit、X-Scan、Hping 这几种工具对靶机发动攻击,产生的攻击包括扫描、漏洞利用、口令猜测、Web 攻击和拒绝服务攻击。

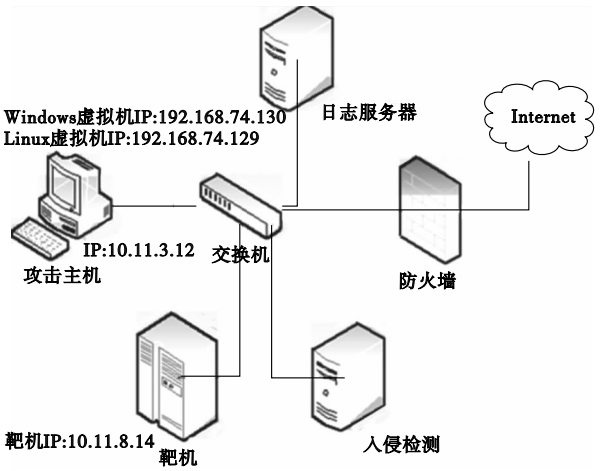

图3　模拟环境的拓扑图

为了衡量告警聚合算法的聚合程度,定义了聚合程度概念,即

$$聚合程度 = \frac{原始告警数 - 超告警数}{原始告警数} \times 100\%$$

图 4 是 Snort 收集的 FTP 口令猜测的原始告警数据。首先对这些告警作预处理，分别解析出告警类型、设备 ID、源 IP、目的 IP、目的 port、开始时间和结束时间，然后再对其进行聚合分析。表 1 为聚合之后的超告警数据。在超告警的属性中，不仅保留原始告警数据属性，还增加了告警个数、告警模式、告警原始 ID 集，从而可以追溯原始告警数据。

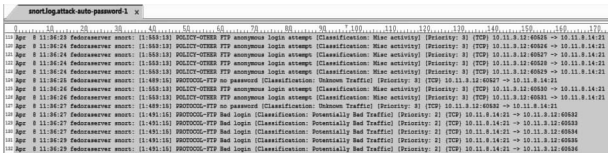

图4 Snort收集的一段原始告警

表 1 聚合之后的超告警数据

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

通过一周的模拟攻击实验，选取任意一段数据作为实验数据。图 5 统计的是几个告警数量相对较多的类型在不同时间区间段的告警数据总量。

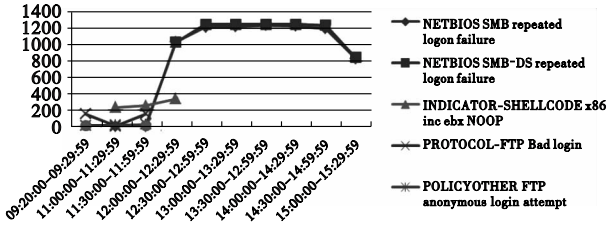

图5 某几种类型在不同时间区间段的告警数据总量

实验数据中 IDS 告警数据 18 232 条，利用本文提出的聚合方法对告警数据进行聚合，聚合结果如图 6 所示。

本文选取了 FTP 相关的三类告警进行聚合分析，得到如图 7 所示的聚合结果。

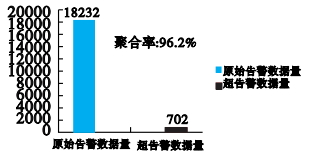

图6 聚合结果

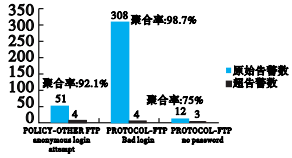

图7 FTP相关的三种类型的告警聚合结果

针对几千条的 NETBIOS 类型相关的告警进行聚合，得到很高的聚合率，如图 8 所示。

为了验证在真实的企业网络环境中本算法的聚合程度，特地采集了某研究所的任意一天的数据来进行聚合分析，得到如图 9 所示的聚合结果。

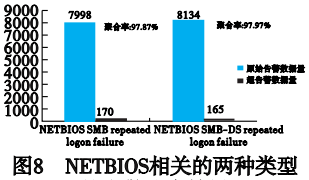

图8 NETBIOS相关的两种类型的告警聚合结果

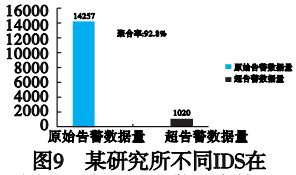

图9 某研究所不同IDS在同一天产生的告警聚合结果

针对本文收集的数据分别用固定时间间隔阈值、文献[5]提出的时间间隔相对均方差阈值聚合和本文提出的动态时间间隔阈值告警聚合三种方法作比较，结果如表 2 所示。

从表 2 实验结果比较来看，在实验数据相同的情况下，本文方法得到了更为精简的超告警，具有更高的聚合率。固定时间阈值和文献[5]的相对均方差阈值方法均需大量的实验以确定一个最佳的阈值。本文的时间间隔阈值是动态更新的，不需要根据专家知识预先确定一个阈值，是一种能够满足不同网络、不同攻击速度的动态调节方法。因此，本文方法更有利于处理实时告警。

表 2 多种聚合方法结果比较

方法	固定时间阈值(时间间隔阈值取 30 s)	文献[5](相对均方差阈值需要事先给定)	本文动态时间阈值
聚合程度/%	87.6 (平均值)	92.3 (平均值)	94.5 (平均值)

4 结束语

本文通过对 IDS、防火墙等安全防御设备的告警数据的分析，提出了一种改进的多源异构告警聚类方法。该方法采用数据融合分析技术，针对告警类型、源 IP、目的 IP、目的端口、时间几个属性进行全面分析，总结出四个规则，并在聚合中动态更新时间间隔阈值。实验结果表明，该方法得到了精简的超告警数据，相比其他方法聚合程度较高。

参考文献：

[1] 郭帆，余敏，叶继华. 一种基于分类和相似度的报警聚合方法[J]. 计算机应用，2007，27(10)：2446-2449.

[2] AI-MAMORY S O, ZHANG Hong-li, ABBAS A R. IDS alarms reduction using data mining[C]//Proc of IEEE World Congress on Computational Intelligence. 2008：3564-3570.

[3] VAARANDI R. Real-time classification of IDS alerts with data mining techniques[C]//Proc of IEEE MILCOM Conference. 2009.

[4] 毛治佳. 基于属性相似度的报警关联系统的研究与实现[D]. 西安：西安电子科技大学，2011.

[5] 龚俭，梅海彬，丁勇，等. 多特征关联的入侵事件冗余消除[J]. 东南大学学报：自然科学版，2005，35(3)：366-371.

[6] CURRY D, DEBAR H. Intrusion detection message exchange format data model and extensible markup language (XML) document type definition. [R]. [S. l.]：Intusion Detection Working Group，2003.

[7] 各种网络攻击模式讲解[EB/OL]. (2012-02-27). http://bbs.csdn.net/topics/380194564.

[8] 李冬. 大规模网络中误告警去除和告警聚类方法研究[D]. 武汉：华中科技大学，2008.

[9] MU Cheng-po, SHUAI Bing. Research on preprocessing technique of alert aggregation[C]//Proc of the 5th International Joint Conference on Computational Sciences and Optimization. 2012：136.