多特征关联的入侵事件冗余消除

龚 俭 梅海彬 丁 勇 魏德昊

(东南大学计算机科学与工程系,南京 210096)
(东南大学江苏省计算机网络技术重点实验室, 南京 210096)

摘要: 通过对事件的源地址、宿地址和宿端口 3 个空间属性进行分析,枚举出事件在空间属性上的所有可能的关联特征;通过对相邻事件的时间间隔进行统计分析,提出了事件的时间关联特征可以用一个相对均方差模型描述.在此基础上给出了一种基于事件类型、空间和时间关联特征的冗余事件消除算法,它能根据冗余消除规则集实时处理入侵事件并进行冗余消除.实验结果表明,该冗余消除算法可以使冗余事件在总的事件中的比例低于 1%,其冗余消除的准确性和消除程度均高于 CITRA 中提出的冗余消除方法.

关键词: 冗余消除;事件关联特征;入侵检测系统;网络安全
中图分类号: TP393 **文献标识码**: A **文章编号**: 1001 - 0505(2005)03-0366-06

Multi-feature correlation redundance elimination of intrusion event

Gong Jian Mei Haibin Ding Yong Wei Dehao

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)
(Key Laboratory of Computer Network Technology of Jiangsu Province, Southeast University, Nanjing 210096, China)

Abstract: All possible correlation features on spatial properties of events are enumerated by analyzing the three spatial properties of events: source address, destination address and destination port. A relative mean square deviation model is proposed by statistical analysis of events' interval, which can be used to describe the temporal correlation features of events. After that, this paper puts forward a redundancy elimination algorithm based on event class, spatial and temporal correlation features, which can deal and eliminate redundant events timely according to the predefined rule set. The experiments show that this algorithm can reduce the rate of redundant events in all events to less than 1%, and is more efficient and accurate than the method used in cooperative intrusion traceback and response architecture (CITRA).

Key words: redundance elimination; event correlation feature; intrusion detection system; network security

入侵检测系统(IDS)是保障网络安全的有效工具,它能够检测网络上的攻击行为,并提示系统管理员进行及时响应,以避免入侵带来的损失[1,2].但目前的 IDS 存在事件报告的冗余现象,即一次攻击产生多个安全事件的报告,这给入侵响应系统准确地自动响应和安全管理员对入侵事件分析造成困扰[3,4].事件报告的冗余来源于 IDS 对持续性攻击的检测,这种攻击会在一段时间内产生多个攻击报文.如果 IDS 不具备对这类攻击起止的识别能力,就会产生重复的事件报告[5].

目前,冗余事件的消除方法主要是基于以下条件:事件的攻击类型相同、事件的源和宿地址相同、事件发生的数量和持续时间阈值.典型的方法有在协同入侵追踪和响应体系结构(CITRA)[6]中提出的冗余事件的"抑制策略"和 IBM 苏黎世研究实验室提出的警报聚集和关联组件(ACC)[7]中的冗余消除算法.CITRA 的方法是确定一个相同攻击事件的数量阈值和时间间隔阈值,超过这些阈值的事件序列就视作不同的事件.显然 CITRA 方法的目的仅在于减少事件的数量,并不关心对持续性攻击

收稿日期: 2004-08-23.
基金项目: 国家自然科学基金资助项目(90104031).
作者简介: 龚 俭(1957—),男,博士,教授,博士生导师,jgong@njnet.edu.cn.

的识别是否准确. IBM 的 ACC 中冗余消除算法的做法是预先定义冗余关系,当有事件到来时,就在以前接收的事件中找和这个事件有冗余关系的初始事件,若找到,就把该事件连接到初始事件上,连接个数超过某个阈值时就处理它,没有考虑事件的时间关系. 因此,它和 CITRA 方法的目的一样仅在于减少事件数量,而不关心识别的正确性. 事实上,并非所有攻击类型对应的冗余事件都要求同源地址且同宿地址,这需要根据攻击类型具体分析. 其次是事件发生的数量和持续时间作为冗余事件的限定需要有适应性,因为一次攻击可能产生的原始事件数量和持续的时间都是不确定的. 因此,这 2 种方法都还比较粗糙,对冗余事件的消除能力有限,甚至还会影响响应的正确性.

本文从攻击类型、空间和时间特征 3 个方面对冗余事件关联特征进行了系统的分析,提出一种基于实时聚类的冗余消除方法,能够更为准确地识别持续性攻击并做出相应的冗余事件消除.

1　事件关联特征分析

1.1　攻击类型关联特征

由于冗余消除的目的是合并一次攻击产生的多个原始事件(IDS 直接输出的事件),因此应当合并的这些原始事件必然具有相同的攻击类型. 但仅凭攻击类型关联特征还无法准确地合并原始事件,因为同一时刻网络上可能有多个并发的同类型攻击,因此还要看它们的时空关联特征.

1.2　空间关联特征

原始事件的空间特征包括以下属性:攻击源地址、攻击源端口、攻击宿地址、攻击宿端口. 由于攻击者在发起攻击时,源端口通常是随机选择的,而其他 3 个属性都有目的性,所以研究事件的空间关联特征主要是针对攻击源地址、攻击宿地址、攻击宿端口. 表 1 列出了所有可能的空间关联特征.

由表 1 可以看出,对于不同类型的攻击共有 9 种不同的空间关联特征,在进行冗余消除时,需要为每类攻击指定空间关联特征.

表 1　原始事件空间关联关系及举例

攻击层次	源地址	宿地址	宿端口	举例
基于传输层以下协议的攻击(宿端口无意义)	相同	相同		Large Ping
	相同	不同		Address Sweep
	不同	相同		Ping Flood
	不同	不同		不存在
基于传输层及以上协议的攻击	相同	相同	相同	Web Attack
	相同	相同	不同	Port Sweep
	相同	不同	相同	Proxy Hunter
	相同	不同	不同	Sweep
	不同	相同	相同	SYN Flood
	不同	相同	不同	UDP Flood
	不同	不同	相同	不存在
	不同	不同	不同	不存在

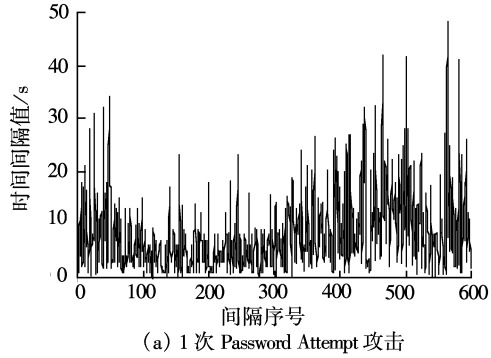
1.3　时间关联特征

由于同类型攻击的持续时间可长可短,而且不同类型攻击的持续时间也有所不同,因此根据一个固定的时间阈值来确定攻击的起止是不合理的. 本文从另一角度来寻找时间关联特征,即原始事件序列 $E_s(e_1, e_2, \cdots, e_n)$ 的相邻事件时间间隔,其中 e_i 表示报告的第 i 个原始事件. 设 t_i 为原始事件 e_i 检测到的时间($i = 1,2,\cdots,n$),又设 $\tau_i = t_{i+1} - t_i$($i = 1,2,\cdots,n-1$),则相邻事件时间间隔为($\tau_1, \tau_2, \cdots, \tau_{n-1}$). 以下实验数据均来自华东(北)地区网络中心开发的网络入侵检测系统 COMON v2 对 CERNET 华东(北)地区网主干信道的检测结果.

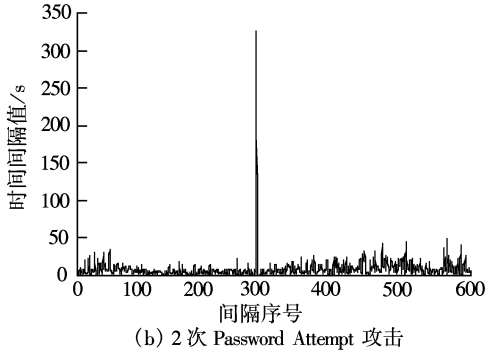
1.3.1　最大时间间隔特征

可以将对一个持续性攻击所检测到的原始事件序列看作一个事件流,因为这些事件是内在相关的. 在现有的流识别方法中,采用固定的时间间隔阈值作为判定依据是一个最常用的方法[8]. 根据这个方法,设原始事件序列 E_s 满足攻击类型关联特征和该攻击类型对应的空间关联特征,则该序列对应同一次攻击,当且仅当 $\tau_i < c(i = 1,2,\cdots,n-1)$,其中 c 为固定的最大时间间隔值.

使用该方法的关键问题是为每种攻击类型确定最大时间间隔值 c. 若 c 值取值过小,则可能不能够充分消除冗余事件. 图1(a)显示了一次Password



(a) 1 次 Password Attempt 攻击



(b) 2 次 Password Attempt 攻击

图 1　最大时间间隔的取值对冗余消除的影响

Attempt 攻击实例,若取阈值为 30 s,则冗余消除结果认为发生了9次该攻击. 若c值取值过大,则可能将多次攻击实例合并. 图 1(b) 显示了 2 次连续的 Password Attempt 攻击实例,若阈值取值 350 s,则这2次攻击实例将被合并为1次攻击实例. 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. 同样的问题在报文流识别中也存在,因此可以考虑更复杂的动态阈值方法,例如 MBET 方法[9].

1.3.2 最大相对均方差特征

从图 1 可以看出,同一次攻击实例的相邻事件时间间隔值有如下特征:它们大多分布在均值附近,即它们相对于均值的偏离较小.

图 2(a) 显示了一次 Large Ping 攻击产生的原始事件序列的各相邻事件时间间隔值,该攻击共产生了 504 个原始事件,共有 503 个时间间隔. 图 2(b) 显示了一次 Password Attempt 攻击,该攻击产生了80个原始事件,共有79个时间间隔值. 对于图 2(a),所有相邻事件时间间隔的平均值为 62.220 7 s. 图 2(b) 中,所有相邻事件时间间隔的平均值为94.137 5 s,其中大部分分布在50 ~ 150 s之间. 这种特征可以通过计算相对均方差(均方差/平均值) 来体现,该值实际刻画了一组数据相对于均值的偏离. 时间间隔相对均方差的计算公式如下:

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

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

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

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

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. 经计算,图2(a) 的相对均方差为0.137 4,图2(b) 的相对均方差为0.463 3. 从结果看,这 2 个攻击对应的相邻原始事件时间间隔值的相对均方差都较小,即时间间隔值的分布集中在均值附近.

产生这种现象的原因在于:一次持续性攻击具有连续性,并且由于这种攻击通常都是由攻击工具自动完成,因此攻击进行的速度比较稳定;另外,由于网络延时、系统响应时间、检测系统局限性等因素的影响,这些时间间隔值相对均值会有小范围的

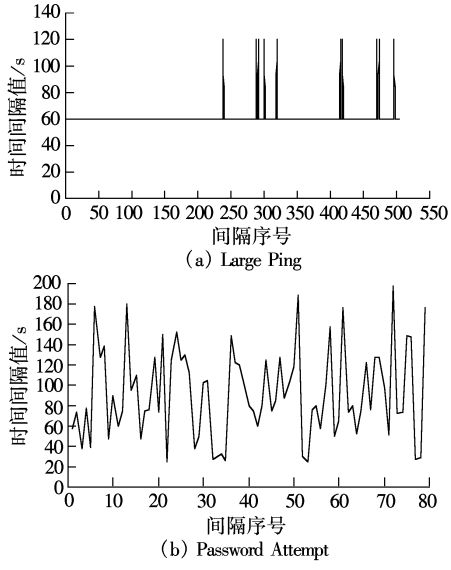


图 2 2 个攻击实例的相邻事件时间间隔值

抖动. 对于图 2(a),Large Ping 攻击持续向目标发送超长 Ping 报文,并且不关心目标系统的响应,因此平均速度非常稳定,大部分时间间隔保持在 60 s左右. 图中少量抖动较大的时间间隔为 120 s 左右,这种现象很可能是由于入侵检测系统在处理高速网络流量时产生的漏报. 对于图 2(b),Password Attempt 攻击需要等待目标系统的响应,由于目标系统响应时间的抖动,使得该攻击的相邻事件时间间隔相对于均值的抖动大于 Large Ping 攻击.

通过对其他攻击实例的研究发现,它们对应的事件都存在这样的时间关联特征. 表 2 列出了其中10 个不同攻击实例的事件时间关联特征. 从表中可以看出这些攻击实例的持续时间、产生事件的数量、产生事件的密度(即相邻事件时间间隔平均值的倒数) 都有所不同,但它们的相邻事件时间间隔值的相对均方差都较小.

因此,冗余事件的时间关联特征可以描述如下:设一次攻击产生的原始事件序列 $E_s, t_i (i = 1,2,\cdots,n)$ 表示 e_i 被检测到的时间,$\tau_i = t_{i+1} - t_i (i = 1,2,\cdots,n-1)$ 表示 e_i 和 e_{i+1} 的时间间隔,则该攻击产生的原始事件的时间关联特征可通过 $\sigma^*(\tau) < c$ 表示,其中 c 为阈值,可以根据每种攻击的不同行为特征指定不同的值,因此是一种随攻击类型变化的动态阈值方法.

使用相对均方差方法的优点在于:它能够更好地适应不同的攻击速度. 该方法根据各时间间隔值相对于平均值的偏离来判断原始事件是否对应同一次攻击,这使得它与攻击速度的快慢无关,因此,可以为每种攻击类型取固定的阈值.

表 2　不同攻击实例的冗余事件时间关联特征

序号	攻击类型	持续时间/s	事件数量	平均值/s	均方差/s	相对均方差
1	Large Ping	432 359	3 944	109.652 3	80.654 3	0.735 5
2	Large Ping	327 427	1 483	220.935 9	205.806 0	0.931 5
3	SubSeven	559	711	0.787 1	0.534 7	0.679 3
4	SubSeven	1 436	2 899	0.495 6	0.405 6	0.818 4
5	Passwd Attempt	14 690	958	15.350 1	10.158 5	0.661 8
6	Passwd Attempt	2 832	498	5.698 2	3.478 7	0.610 5
7	NMAP TCP Ping	13 374	4 388	3.048 6	2.896 0	0.950 0
8	NMAP TCP Ping	3 196	1 059	3.020 8	3.271 2	1.082 9
9	Ingreslock	55 988	370	151.729 0	147.517 4	0.972 2
10	Ingreslock	57 535	322	179.236 8	172.857 4	0.964 4

2　冗余消除算法

如果采用具有动态阈值的冗余消除方法,需要针对不同的攻击类型定义冗余消除规则,以描述事件的冗余标准. 在 COMON v2 的实现中,其语法的 Backus 表示法如下:

〈冗余消除规则〉::=〈AC〉 and 〈SC〉 and 〈TC〉

〈AC〉::= attack =〈攻击类型名称〉

〈SC〉::= same = ‘{‘〈属性表〉’}’

〈TC〉::= maxv =〈数值〉

〈属性表〉::=〈属性〉{,〈属性〉}

〈属性〉::= SRCIP | DSTIP | DSTPORT

其中,AC 约束指明了攻击类型;SC 约束指明了被合并的原始事件之间必须有相同的属性域,可选的属性包括 SRCIP(攻击源地址)、DSTIP(攻击宿地址)、DSTPORT(攻击宿端口);TC 约束指明了被合并的原始事件的相邻事件时间间隔值的相对均方差的最大值. 在冗余消除规则脚本中,每种可能产生冗余事件的攻击类型对应惟一的一条冗余消除规则,每条冗余消除规则由〈AC〉约束中定义的攻击类型惟一标识.

冗余消除算法独立于入侵检测过程,它的输入是入侵检测模块输出的原始事件流. 算法的基本思想是:为每一个正在进行的且可能产生冗余事件的攻击保存一个状态 S(称为简单攻击状态),该状态包含该攻击到目前为止产生的所有原始事件的相关信息. 每收到一个原始事件都进行冗余判定,在调整相应的 S 的同时丢弃冗余的原始事件. 当判定攻击结束时,根据对应 S 的当前信息生成一个简单攻击事件. 这个思路与用于报文流识别的 MBET 方法的实现思路相似. 对于无后续事件到达的状态,系统设置一个计时器,按最大时间间隔特征定期检查当前的所有状态 S,结束那些已经超时的状态.

设原始事件 e 描述为多元组形式:(ATTACK_TYPE, SRCIP, DSTIP, DSTPORT, DTIME). 其中,ATTACK_TYPE 指攻击类型;SRCIP 指攻击源 IP 地址;DSTIP 指攻击宿 IP 地址;DSTPORT 指攻击宿端口;DTIME 指事件被检测到的时间.

简单攻击事件 SA 也描述为多元组形式:(ATTACK_TYPE, SRCIP, DSTIP, DSTPORT, START_TIME, END_TIME, ATTACK_NUM). 其中,ATTACK_TYPE 表示简单攻击的类型;SRCIP,DSTIP,DSTPORT 和原始事件中的属性含义相同,但这 3 个属性的值都为集合;START_TIME 表示简单攻击的开始时间;END_TIME 表示简单攻击的结束时间;ATTACK_NUM 表示该简单攻击中所包含的原始事件的个数.

在算法中,如果按式(1)~(4)来计算序列中时间间隔的相对均方差,则需要:①保留所有的事件间隔时间 τ_i;②每次新事件到来后,需要利用序列中所有的 τ_i 重新计算平均值、方差和相对方差. 显然,无论从空间复杂度还是时间复杂度上看,都不可取. 经过数学分析后发现,计算过程可以简化,具体过程如下.

设老序列为$(\tau_1, \tau_2, \cdots, \tau_{n-1})$,其中 n 为序列中原始事件个数,则当新一个事件 e^* 到来后,构成的新序列为$(\tau_1, \tau_2, \cdots, \tau_{n-1}, \tau_n)$. 其中 τ_n 为新来事件 e^* 与老序列中最后一个事件的时间间隔,令

$$A = \sum \tau_i = \tau_1 + \tau_2 + \cdots + \tau_{n-1}$$

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

$$A' = \sum \tau_i + \tau_n = A + \tau_n$$

$$B' = \sum \tau_i^2 + \tau_n^2 = B + \tau_n^2$$

则新序列的平均值为

$$\tau'_{\text{avg}} = \frac{A'}{n} \tag{5}$$

方差为

$$\sigma(\tau) = \sqrt{\frac{1}{n}\left(B' - \frac{A'^2}{n}\right)} \tag{6}$$

相对均方差为

$$\sigma^*(\tau) = \frac{\sigma(\tau)}{\tau'_{\text{avg}}} \tag{7}$$

这样,只要保留序列的 A,B 和 n,就可利用新来事件的时间间隔算出新序列的相对均方差.

设 S 表示为多元组形式:(ATTACK_TYPE,

SRCIP,DSTIP, DSTPORT,T_s,T_e,n,A,B). 其中,T_s 为序列的第 1 个原始事件被检测到的时间;T_e 为序列最近接收的原始事件被检测到的时间;n 为序列中原始事件的个数;A 为所有原始事件相邻时间间隔和;B 为所有原始事件相邻时间间隔平方和,其他部分与 SA 的属性含义相同.

用双向链表 T 保存当前简单攻击状态的集合. 算法初始时,设置可能的最大事件间隔值 INITIAL_MAX_INTERVAL,使 T = null,并启动定时器 TimerA,则基于实时聚类的冗余消除算法的主要流程如下.

① 接收原始事件(设为 e^*),根据 e^*. ATTACK_TYPE 查找相应的冗余消除规则 R,即找规则 R 中定义的攻击类型和 e^* 的攻击类型一样的,找到则转②;否则,将该事件直接报送,然后转①.

② 搜索存放简单攻击状态的双向链表 T,找简单攻击状态 S',判断 S' 和 e^* 能否满足 R 规则的攻击类型约束、空间约束和时间约束,不满足 R 规则的攻击类型或空间属性约束或二者都不满足,则转④;否则,计算 σ^*, 过程为

$$n = S'.n + 1,\ \tau_n = e^*.\mathrm{DTIME} - S'.T_e$$

$$A = S'.A + \tau_n,\quad B = S'.B + \tau_n^2$$

$$\tau_{avg} = \frac{A}{n},\quad \sigma = \sqrt{\frac{1}{n}\left(B - \frac{A^2}{n}\right)},\quad \sigma^* = \frac{\sigma}{\tau_{avg}}$$

若 $\sigma^* < c$(c 为 R 规则中设定的相对均方差阈值),则说明全部满足 R 的条件,则转 ③;若 $\sigma^* > c$,则不满足时间约束,说明 S' 超时,则产生一个简单攻击事件SA并报送,然后将S'状态从T中删除,转 ④.

③ 更新简单攻击状态 S',令

$$S'.A = S'.A + \tau_n,\quad S'.B = S'.B + \tau_n{}^2$$

$$S'.n = S'.n + 1,\quad S'.T_e = e^*.\mathrm{DTIME}$$

$$S'.\mathrm{SRCIP} = S'.\mathrm{SRCIP} \cup e^*.\mathrm{SRCIP}$$

$$S'.\mathrm{DSTIP} = S'.\mathrm{DSTIP} \cup e^*.\mathrm{DSTIP}$$

$$S'.\mathrm{DSTPORT} = S'.\mathrm{DSTPORT} \cup e^*.\mathrm{DSTPORT}$$

其中,$\tau_n = e^*.\mathrm{DTIME} - S'.T_e$,转 ⑤.

④ 创建新的简单攻击状态 S_{new},令

$$S_{new}.T_s = e^*.\mathrm{DTIME},\quad S_{new}.n = 1$$

$$S_{new}.T_e = e^*.\mathrm{DTIME}$$

$$S_{new}.A = 0,\ S_{new}.B = 0$$

$$S_{new}.\mathrm{SRCIP} = S_{new}.\mathrm{SRCIP} \cup e^*.\mathrm{SRCIP}$$

$$S_{new}.\mathrm{DSTIP} = S_{new}.\mathrm{DSTIP} \cup e^*.\mathrm{DSTIP}$$

$$S_{new}.\mathrm{DSTPORT} = S_{new}.\mathrm{DSTPORT} \cup e^*.\mathrm{DSTPORT}$$

转 ⑥.

⑤ 依次遍历简单攻击状态双向链表 T 中的简单攻击状态 S,对每个简单攻击状态 S,cur_time 为当前系统时间,令 τ_n = cur_time − $S.T_e$, 若 τ_n 大于设定的时间间隔时间 INITIAL_MAX_INTERVAL,则说明 S 超时,就产生一个简单攻击事件 SA,报送 SA,然后将 S 状态从 T 中删除. 该步骤重复直至不存在超时的状态,转 ①.

⑥ 检查定时器 TimerA,时间还没到则转 ①,否则,重新启动定时器,转 ⑤.

算法中产生简单攻击事件 SA 的方法是:SA.ATTACK_TYPE = S.ATTACK_TYPE,SA.SRCIP = S.SRCIP,SA.SRCPORT = S.SRCPORT,SA.DSTIP = S.DSTIP,SA.START_TIME = $S.T_s$,SA.END_TIME = $S.T_e$, SA.ATTACK_NUM = $S.n$.

3 实验与优化

为验证冗余算法的冗余消除性能,本文采用了 COMON v2 在网上收集的 2 次原始事件数据文件作为输入,分别用本文方法和 CITRA 方法对它们进行冗余消除. 冗余算法从文件中读取这些原始数据,根据冗余消除规则进行冗余消除处理后,把输出的简单攻击事件存放在简单攻击事件文件中. 为了衡量冗余消除程度,本文引入冗余程度概念,即冗余程度 =

$$\frac{\text{原始事件数量} - \text{输出的简单攻击事件数量}}{\text{原始事件数量}} \times 100\%$$

对于数据 1,其中原始事件数为 500 001 条,事件种类数为 85 种,用本方法冗余消除后输出的简单事件数量为 17 284 条,冗余程度为 96.54%;用 CITRA 方法输出的简单事件数量为 23 345 条,冗余程度为 95.33%. 对于数据 2,其中原始事件有 50 001条,事件种类数为 53 种,本方法冗余消除后输出的简单事件数量为 1 605 条,冗余程度为 96.79%;用 CITRA 方法输出的简单事件数量为 2 477条,冗余程度为 95.05%.

为了检查本文方法是否存在优化问题,对数据 1 的原始文件和输出的结果文件均导入到 SQL Server 2000 数据库中,并进行了统计分析,表 3 是统计出的发生次数多且冗余程度偏小的前 10 种基于 SNORT 规则的攻击事件.

把表 3 中的每种事件分别通过手工方法从输出的简单攻击事件结果中查询出来进行验证,发现序号为 2,5,7 和 9 的事件还存在冗余,其他事件基本没有冗余. 通过查看冗余事件是空间还是时间上存在冗余,和查看相应的冗余规则,发现冗余的原因是冗余规则不够完善. 修改对应的冗余规则后,再次运行冗余消除算法,则输出结果如表 4 所示,并且输出的简单攻击事件已不再存在冗余.

表 3　冗余程度偏小的前 10 种事件

序号	事件描述	原始事件数量	简单事件数量	冗余程度/%
1	WEB-MISC RBS ISP/newuser access	1 088	846	22.24
2	WEB-MISC http directory traversal	9 647	5 511	42.87
3	DOS real audio server	6 025	3 139	47.90
4	WEB-IIS scripts access	7 478	1 489	80.09
5	SNMP trap udp	1 090	158	85.50
6	WEB-MISC login. htm access	1 964	189	90.38
7	SNMP request udp	2 890	249	91.38
8	ICMP destination unreachable	10 671	918	91.40
9	ICMP source quench	2 547	103	95.96
10	WEB-PHP phpBB privmsg. php access	2 040	53	97.40

表 4　修改规则后 4 种事件冗余消除结果

序号	事件描述	原始事件数量	简单事件数量	冗余程度/%
2	WEB-MISC http directory traversal	9 647	369	96.17
5	SNMP trap udp	1 090	108	90.09
7	SNMP request udp	2 890	98	96.61
9	ICMP source quench	2 547	66	97.41

因此,本方法的优化工作应是对数据进行统计分析,找到合理的冗余消除规则. 最后,经过规则优化后本方法的运行结果为:对数据 1,输出的简单事件数量为 11 904 条,冗余程度为 97.62%;对于数据 2,输出的简单事件数量为 1 071 条,冗余程度为 97.86%.

最后将冗余消除后产生的简单攻击事件作为 COMON v2 自动响应功能中[5,10]宏观攻击行为分析模块的输入,进行挖掘分析,分析结果表明,简单攻击事件集合中的可能的冗余事件比例低于 1%,并且发现它们大多对应于慢速攻击.

在冗余消除的准确性上,本文方法和 CITRA 方法也做了比较. 限于篇幅,本文仅举 1 个例子. 在原始事件数据中,有源 IP 为 XXX. XX. XX. XXX(考虑到隐私问题,已将 IP 地址做了净化处理),时间从 2004-07-15T23:54:01 到 2004-07-16T00:03:05,且事件时间间隔均不超过 5 s 的40 188次 ICMP Ping 事件. 显然这些原始事件可以合并为一次简单攻击事件. 经过本方法冗余消除后输出确为一次简单攻击事件,而 CITRA 方法输出为 29 次简单攻击事件. 可以看出本方法比 CITRA 方法冗余消除的准确性高.

以上分析和实验结果表明,本文提出的冗余消除算法能够根据用户配置的冗余消除规则很好地消除原始事件中的冗余,并且在冗余消除程度和准确性上均优于 CITRA 方法.

4　结　语

目前国内外的入侵检测系统,都存在大量冗余事件的问题,本文通过对入侵事件的分析,提出了一种利用事件的多特征关联进行冗余事件消除的新方法. 特别是对事件的时间关联特征上,本文提出了相对均方差模型,对事件数据的分析表明,该模型能精确地刻画事件的时间关联特征. 在提取了事件多特征关联之后,本文给出了基于实时聚类的冗余消除算法. 实验证明本算法可以很好地消除冗余事件,并在冗余消除程度和准确性上要优于 CITRA 方法.

作为本文的后续工作,在冗余消除规则的制定上还有待研究. 目前冗余消除规则主要靠手工制定,工作量大且不够灵活,以后将研究利用数据挖掘的方法来自动寻找冗余消除规则.

参考文献(References)

[1] 龚　俭, 陆　晟, 王　倩. 计算机网络安全导论[M]. 南京:东南大学出版社, 2000. 245 - 247.

[2] Mukherjee B, Heberlein L T, Levitt K N. Network intrusion detection [J]. *IEEE Network*, 1994, **8**(3):26 - 41.

[3] Julisch Klaus. Clustering intrusion detection alarms to support root cause analysis[J]. *ACM Transactions on Information and System Security*, 2003,**6**(4):443 - 471.

[4] Ning Peng, Cui Yun, Reeves Douglas, et al. Tools and techniques for analyzing intrusion alerts [J]. *ACM Transactions on Information and System Security*, 2004, **7**(2):273 - 318.

[5] 丁　勇. 自动入侵响应系统的研究[D]. 南京:东南大学计算机科学与工程系, 2004.

[6] Schnackenberg Dan, Holliday Harley, Smith Randall, et al. Cooperative intrusion traceback and response architecture(CITRA)[A]. In: *Proceedings of the Second DPRPA Information Survivability Conference and Exposition* [C]. Anaheim, CA, 2001, **1**:56 - 68.

[7] Debar H, Wespi A. Aggregation and correlation of intrusion-detection alerts[A]. In: *Proceedings of the 4th Symposium on Recent Advance in Intrusion Detection (RAID)*, *LNCS*[C]. Berlin: Springer Verlag, 2001. 85 - 103.

[8] Claffy K C. Internet traffic characterization [D]. San Diego: University of California, 1994.

[9] Ryu B, Cheney D, Braun H W. Internet flow characterization: adaptive timeout strategy and statistical modeling[A]. In: *Proceedings of Passive and Active Measurement Workshop*[C]. Amsterdam, 2001. 94 - 105.

[10] 张　剑. 可回卷的动态反馈自动入侵响应系统[D]. 南京:东南大学计算机科学与工程系, 2004.