入侵检测系统的冗余消除方法综述[1]

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

摘要：随着网络规模的扩大，网络速度的提高，IDS 所检测到的事件数量急剧上升，由于 IDS 本身的局限，这些安全事件中难免会存在冗余，冗余事件对于系统监测和入侵响应均有不利影响。本文介绍几种冗余消除方法，并比较它们的优劣。

关键字： IDS 冗余消除

1. 引言

近年来随着计算机网络规模和应用的不断扩大，安全事件的数量急剧上升，攻击技术更加复杂，造成的危害也愈加严重。因此出现了很多安全技术用于增强网络的安全性，入侵检测系统（Intrusion Detection System）就是其中之一，它能够检测网络上的攻击行为，期望在入侵行为造成危害前实时发出警报，并采取相关的措施以避免入侵带来的损失。

根据检测方法的不同，入侵检测技术科分为异常检测（Abnormal Detection）和滥用检测（Misuse Detection）。异常检测通过采集和统计发现网络或系统中的异常行为，然后按照某种决策判断它是否是入侵行为，它可以检测出未知的攻击方式，但误报率较高；滥用检测依赖已知的知识建立各种攻击模式，对比采集的信息来判定行为是否是入侵行为。尽管两种检测方法并不相同，但它们面临同样的问题：对同一个持续攻击行为 IDS 可能会产生重复的报警事件，称之为冗余事件。对于低速小规模网络而言，冗余事件数量较少，IDS 可以并不需要有效的冗余消除，但对于大规模高速网络的 IDS，数量庞大的冗余事件有着极其不利的影响。一些商业的 IDS 采取了相关措施解决冗余事件的问题，如 McAfee 的 Intrushield 2600 可以通过事件关联，将与一个入侵相关的成百上千的事件归并到一个入侵事件中；绿盟科技的冰之眼根据一系列事先制定的策略，将多条警告日志合并为一条。

冗余事件的数量与检测的粒度、网络的规模有直接关系。大量的冗余事件会使IDS对同一攻击做出很多重复的不必要的响应，降低系统的性能和效率，甚至被攻击者用于攻击IDS，使其无法响应过多的事件而服务失效。此外，冗余事件还会给管理员对安全事件的查询造成了困难，特别是这些冗余事件有成千上万条时，将使得IDS的输出不可读。虽然冗余消除问题可以在入侵检测算法中解决：在进行入侵检测时，为每个持续性攻击保存状态，从而为每个攻击只产生一个安全事件。但过多的中间状态将消耗系统资源，增加检测算法的复杂度，降低检测算法的效率，而且不利于检测算法的简化和优化。因此，在IDS中，特别是面向大规模高速网络的IDS中，冗余消除是必不可少的功能模块。冗余消除的难点在于如何合理精确地消除冗余事件，使检测结果尽可能与攻击行为一一对应，从而提高响应的准确性。

本文主要介绍现有的几种冗余消除的方法，并比较它们的优劣。

2. AC 的方法

2001 年，法国电信研发中心的 H. Debar 和 IBM 苏黎世研究实验室的 A. Wespi 提出了警

[1]本文受国家自然科学基金课题 90104031 资助.

报聚集和关联算法(the Aggregation and Correlation Algorithm，简称 AC 算法)[1]，该算法用于检测入侵事件并关联它们的意图，使最终的报警事件更具体。在分析事件间的关系时算法定义了两种事件间的关系：关联关系和聚集关系。

冗余（duplicate）是关联关系的一种，可以用一个四元组描述：（初始事件类型，冗余事件类型，属性列表，安全等级）。初始事件是指检测到的未经处理的事件；冗余事件类型取决于初始事件的类型；属性列表是冗余事件的相同属性的集合；安全等级用于量化冗余事件的危害，用于进一步的处理。

当接收到一个事件时，首先搜索当前事件的冗余关系的描述，然后通过匹配冗余属性在以前接收的事件中搜索与该事件有冗余关系的初始事件，如果找到，就把这个事件连接到初始事件上。当事件数超过某个阀值时才处理它。

这种方法对冗余事件的定义比较详细，能够比较精确地判定新到来的事件是否与原有的事件冗余。但是因为攻击持续的时间并不固定，产生的冗余事件个数也不确定，所以仅用事件阀值并不能正确地识别攻击，而且阀值的设定还有很多问题需要考虑，因为没有超时机制，如果阀值设置过大，将会导致漏报，甚至可能将不同时间段的攻击当作一次攻击；反之，过小的阀值则不能够有效降低事件的冗余度。这种方法只能降低冗余事件的数量，并不能正确有效地识别攻击。

3．CITRA 的方法

同年，波音公司的Dan Schnackenberg和NAI实验室的Kelly Djahandari等人共同发表了一篇文章，提出了协同入侵追踪和响应体系结构（Cooperative Intrusive Traceback and Response Architecture 简称CITRA）[2]。该结构将入侵检测系统、防火墙和路由器结合在一起追踪攻击的真实源点，并且阻断攻击源。它还能够与其它的安全机制相互交户，对入侵和其他一些系统安全状态的变化自动做出响应。

在 CITRA 的设计中针对冗余消除问题提出了“抑制策略（throttle policy）”的概念。“抑制策略”主要定义了两个参数：事件数量阀值、时间范围阀值，当检测到的同一类型的事件超过某个阀值时，CITRA 才将该事件上报。

图 1 是一条 CITRA 的抑制政策，它表示：如果 CITRA 检测到的端口扫描（PORT_SCAN）事件累积到 20 条（即事件数量阀值，由 threshholdnum 指定），或者定时器超过了 20 秒（即时间范围阀值，由 threshholdtime 指定）时，则 CITRA 报告一个该类型的安全事件。

```
[THROTTLE]
    [THROTTLE/1]
        attack_code = PORT_SCAN
        threshholdnum = 20
        threshholdtime = 20
```

图 1 CITRA 抑制策略的示例

但这种方法存在一些不足。在抑制策略中事件类型定义模糊，如冗余的端口扫描事件的源地址和目标地址相同，但其他的攻击类型（如 DDOS 攻击）的冗余事件未必要求相同的源地址和目标地址，而这种区别在抑制策略的攻击类型字段（attack_code）中是无法表现的。此外事件数量阀值和时间范围阀值这两个参数使用绝对的数值并不合理。因为一次攻击可能产生的原始事件数量是不确定的，如端口扫描攻击根据扫描范围的大小，产生的原始事件数量有多有少；一次攻击可能持续的时间也是不确定的，攻击持续的时间和攻击类型、网络速度、攻击者都有关系。

该方法能够一定程度地降低事件的冗余程度，在最好情况下能常数倍降低冗余事件的数量，如上例最大的冗余程度为 20，所以冗余消除的效果并不理想。

4. 多特征关联的冗余消除方法

2004 年，东南大学的丁勇在他的硕士论文[3]中通过分析冗余事件的多个关联特征，提出了一种基于实时聚类的冗余消除方法。冗余事件的关联特征包括：攻击关联特征，空间关联特征和时间关联特征。某一事件与它的冗余事件必须拥有相同的攻击类型，如果不相同，则表示它们是由不同的攻击所造成的，它们之间不可能有冗余关系。攻击的源端口通常是随机选择的，因此空间关联特征仅包括：源地址，目标地址和目标端口。每种攻击类型的空间关联特征需要专门指定，表 1 列出了所有可能的空间特征[3]。时间关联特征用于判定某次攻击是否结束。

表 1 原始事件空间关联关系及举例

源地址	宿地址	宿端口	举例	攻击所处的层次
相同	相同	X	Large Ping	基于传输层以下协议的攻击（宿端口无意义）
相同	不同	X	Address Sweep	
不同	相同	X	Ping Flood	
不同	不同	X	不存在	
相同	相同	相同	Web Attack	基于传输层及以上协议的攻击
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	

该算法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，在进行流识别即判定攻击起止时，并没有采用通常使用的固定时间阀值的做法，而是从原始事件序列的相邻事件的时间间隔考虑，动态地确定时间阀值。文章通过对大量的攻击实例的相邻事件时间间隔值的统计，得出一个结论：同一次攻击实例的相邻事件时间间隔值大多分布在均值附近，即它们相对于均值的偏离较小。因此该算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。当新的事件到达时，可以得到它与前一次冗余事件的时间间隔，从而计算出该次攻击已产生的所有事件时间间隔的相对均方差，如果不大于给定的阀值，则视为攻击仍在持续，否则认定原攻击已经结束，该事件是由新攻击的产生的。

冗余消除过程独立于入侵检测过程，算法实时地接收入侵检测获得的原始事件，将事件流中对应于同一次攻击的多个原始事件合并成一个简单攻击事件。该算法的基本思想是：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（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通过匹配关联特征进行冗余判定，调整相应的简单攻击状态，同时丢弃冗余的原始事件。当判定攻击结束时，根据对应简单攻击状态的当前信息生成一个简单攻击事件。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，结束那些已经超时的状态。

这种多关联特征的算法对冗余事件特性的描述比较详细，特别是时间关联特征采用相对均方差阀值，可以使算法与攻击速度的快慢无关，能够更好地适应不同的攻击速度。该算法对攻击识别的结果比较精确，冗余消除的效果也比较好。但它仍存在一定的问题，如阀值的确定。即使是使用相对均方差的方法，仍然要求设定一个阀值。为了保证攻击识别的准确性，必须为每种攻击类型设定阀值，而且较为理想的阀值还需要经过一定时间的调整。

5. 其他的方法

2001年，法国ONERA的F. Cuppens在文献[4]中，为解决多个IDS所产生的冗余事件，提出了一种事件聚类（Alert Clustering）的方法，通过定义相似关系（Similar Relation）对事件进行聚类。相似关系包括分类相似性，时间相似性，攻击源和目标的相似性。分类相似性用于定义攻击类型，使它不受不同IDS的影响；攻击源和目标一般可由节点（IP地址或主机名表示）、用户、服务和进程描述，如果某个具体攻击所要求的描述全部相似，则认为是满足攻击源和目标的相似性；如果两个事件被检测的时间间隔大于阀值，则认为两个事件满足时间相似关系。满足相似关系的事件将被聚为一类。这种方法对冗余事件作了详细的描述，时间阀值的设置采用的是事件间的最大时间间隔，但它没有设置事件阀值，在处理突发的大量事件时可能产生误报。

一些公司也开发出了商业用途的 IDS，它们也使用了一些冗余消除的方法。如联想公司的网御 N200，它采用事件压缩再分析技术（Event Merge Analysis，EMA）对报警事件进行缩略合并。事件缩略再分析模块依据预设的规则，对相同的一个源地址或目的地址的数据进行分析，对在定义的时间段内发生的相同类型的攻击视为同一攻击事件[5]。例如对于端口扫描攻击事件，事件缩略再分析模块针对具有统一的目的地址、源地址和不同端口号的数据包分析，并判断出其为扫描攻击，合并为一次事件上报。这种方法与 CITRA 的方法类似，冗余事件的定义不够精确，而且时间阀值采用最大阀值，不同之处在于它并没有设置事件阀值，这可能导致对突发事件的冗余消除率过大。

6. 总结

入侵检测输出的原始事件可能存在冗余，正确地识别出攻击对 IDS 的监测和响应意义重大。上述几种冗余消除的方法各有优劣。这几种方法都对冗余事件作了定义，只是精确程度略有不同。并且都使用阀值限定事件的合并：事件阀值可以合并一定数量的事件，但当只有少量事件时可能合并不同时间攻击所产生的事件；时间阀值可以合并一定时间段内的冗余事件，但对突发的大量事件可能会合并不同攻击所产生的事件。因此设置两种阀值协同作用效果较好。阀值的设置方法也不相同：最大阀值的做法较为简单，但无法为每种攻击类型确定固定的阈值，攻击的速度受到多种因素的影响，冗余事件产生的速度不同，冗余消除可能不够精确；利用均方差确定阀值的做法虽然能够不受攻击速度的影响，但较为复杂。以上几种方法仅适用于滥用检测方式，对于未知攻击，它们无法对冗余事件做出定义或描述，因此很难适用。攻击识别的精度还有待进一步提高。

参考文献

1. H. Debar, A. Wespi; Aggregation and Correlation of Intrusion-Detection Alerts; In Proceedings of the 4th

International Symposium, Recent Advances in Intrusion Detection (RAID) 2001

2. Dan Schnackenberg, Harley Holliday, Randall Smith, et al; Cooperative Intrusion Traceback and Response Architecture(CITRA); Proceedings of the second DARPA Information Survivability Conference and Exposition; June, 2001
3. 丁勇，龚俭；自动入侵响应系统的研究；东南大学硕士学位论文；2004.05
4. F. Cuppens; Managing Alerts in a Multi-Intrusion Detection Environment; 17th Annual Computer Security Applications Conference (ACSAC). New-Orleans, December 2001
5. 王亚平；联想网御 IDS 两个新焦点；中国计算机报；2003，71 期