# 基于报警原因的聚类分析方法

王秀利 海 沫 朱建明 章 宁
(中央财经大学信息学院 北京 100081)

**摘 要** 针对入侵检测系统产生大量冗余报警的问题,提出基于报警原因的聚类分析方法。根据报警原因把逻辑上相关的报警归类到同一个报警聚类中,聚类中的报警具有相同的属性,进而归纳为泛化报警,并由它描述报警的共同特征,从而极大地减少报警数量,简化报警分析,有利于准确分析出网络和应用环境面临的安全威胁,以及时采取应对措施。

**关键词** 入侵检测,报警分析,报警聚类,报警原因,启发式算法
**中图法分类号** TP393.08 **文献标识码** A

## Clustering Analysis Method Based on Alert Cause

WANG Xiu-li HAI Mo ZHU Jian-ming ZHANG Ning
(School of Information, Central University of Finance and Economics, Beijing 100081, China)

**Abstract** The use of intrusion detection has created the problem to investigate a generally large number of alarms. To solve the problem, a clustering analysis method based on alert cause was presented. The correlative alarms with the same attribute were ranged into a clustering according to their causes. The generalized attributes can describe the common characteristic of the alarms. The method can cut down the number of alarms remarkably, simplify the alert analysis, and analyze the security risk in network and application environment accurately. Therefore, the corresponding measures can be taken in time.

**Keywords** Intrusion detection, Alert analysis, Alert clustering, Alert cause, Heuristic algorithm

## 1 引言

现有的网络入侵检测系统侧重于低层次的检测功能,没有对检测结果进行进一步分析。由于报警数据中存在大量的冗余信息,使得报警数据难以理解和使用。因此,需要对报警数据进行分析,在孤立的报警之间寻找逻辑关联,排除冗余报警,进而发现系统的安全隐患。

为此,研究人员提出了多种报警分析方法。文献[1]基于报警属性的相似性建立报警间的可能关系,对有相同源地址和目的地址的报警可以有效建立报警间的关系,但是,这种方法不能发现相关报警间的因果关系。因此,不能从报警中探寻攻击的前提条件和后果并建立攻击场景。文献[2,3]根据事先指定的攻击场景或训练数据库分析报警,这种方法局限于已知的攻击场景,不能有效发现新的攻击方法或攻击步骤导致的报警间的关系。文献[4-6]基于攻击的前置条件和结果建立攻击场景,把孤立的报警关联起来,从静态报警中构建动态的攻击行为,通过连续发生的报警事件重建攻击过程。文献[7]基于单个攻击的前提和结果,认为如果一个报警的前提条件满足于另一个报警的结果,则两个报警间存在关联,这类方法能发现报警信息间的因果关系,并且不受已知攻击场景的限制。文献[8]通过分析规则来分析报警,在报警和规则间建立联系,通过标识报警和对应规则之间的关系来处理报警。文献[9]通过分析过去的报警研究将来可能出现的报警。这种方法在稳定的网络环境中具有较高的分析效率,能快速有效地发现曾经出现的攻击行为和方式,但对新的攻击类型和方式的发现具有一定的滞后性。

本文的贡献在于提出一种基于报警原因的聚类分析方法。通常情况下,一次恶意行为就会触发多个规则,并导致多次报警。逻辑上这些报警由同一恶意行为引起,它们有相同的报警原因。而且同一恶意行为可能多次发生,导致类似报警被记录多次。在入侵检测系统运行过程中,虽然报警数量很大,但报警原因相对较少。因此,本文根据报警原因把逻辑上相关的报警归类到同一个报警聚类中,进而归纳为泛化报警,以极大地减少报警数量,简化报警分析。

## 2 基本概念

本节首先描述报警原因、报警原因分析、报警聚类和泛化关系等几个基本概念。

到稿日期:2009-06-03 返修日期:2009-08-18 本文受国家自然科学基金项目(60970143,70872120,70872119),教育部科学技术研究重点项目(109016),北京市自然科学基金项目(9092014,4082028),北京市教育委员会共建项目专项,中央财经大学"211工程"三期重点学科建设项目,中央财经大学"中财121人才工程"青年博士发展基金项目(QBG0702)资助。

王秀利(1977—),男,博士,讲师,CCF会员,主要研究方向为网络安全、可信计算等,E-mail:xlwang.cufe@gmail.com;海 沫 博士,讲师;朱建明 博士,教授;章 宁 博士,副教授。

报警原因是触发报警的根本原因，是影响网络和应用环境安全并最终导致报警产生的安全问题。如 TCP/IP 协议栈实现中的缺陷就是一个报警原因，它导致“碎片 IP 数据包”报警；蠕虫也是一种报警原因，它影响了感染的主机，并在蠕虫传播时引起报警。

报警原因分析是发现报警原因及其对网络和应用环境产生影响的过程。入侵检测系统记录违背安全策略的事件。报警通过报警模型 $dom(A_1) \times \cdots \times dom(A_n)$ 笛卡尔积表示，记作 $\times_{1\leqslant i\leqslant n} dom(A_i)$，其中 $\{A_1 \cdots A_n\}$ 表示报警的属性，$dom(A_i)$ 为报警属性 $A_i$ 的取值，$Dom(A_i)$ 为属性 $A_i$ 的值域。$A_i$ 反映了报警的一个方面，如源地址、目的地址、报警类型等。

报警聚类是具有公共特征的同一类报警的集合。泛化报警是对报警聚类中报警的归纳和抽象，它包含报警聚类的共同特征，并在一定程度上说明产生该类报警的原因。

泛化关系是报警属性之间的包含关系。如果报警属性 $A_i$ 的取值是另一个报警中对应属性 $A_i$ 的子集，则它们之间是泛化关系。图 1 显示了 IP 地址的泛化关系，这种泛化关系图通常为单根结构的有向无环图。

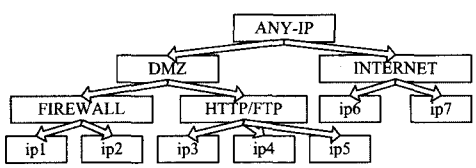


图 1　IP 地址泛化关系图

对于两个元素 $a, \tilde{a} \in Dom(A_i)$，如果在有向无环图中存在从 $\tilde{a}$ 指向 $a$ 的路径，则 $\tilde{a}$ 包含了 $a$，泛化关系图中 $\tilde{a}$ 为 $a$ 的父节点，它们之间的关系表示为 $\tilde{a} \nabla a$。这种关系可以扩展到报警之间的关系，设报警 $o, \tilde{o} \in \times_{1\leqslant i\leqslant n} Dom(A_i)$，当且仅当对所有的属性 $A_i$ 来说，$\tilde{o}[A_i] \nabla o[A_i]$ 时，则 $\tilde{o}$ 包含 $o$。如果 $\tilde{o} \nabla o$，则报警 $o$ 是 $\tilde{o}$ 的特例，也就是说 $\tilde{o}$ 表示的报警更一般。

假设报警原因为“有缺陷的 TCP/IP 协议栈”，这个协议栈发出碎片化的数据包，从而导致“碎片 IP 数据包”报警，并且这个协议栈位于工作日运行的 Web 服务器上。因此，网络入侵检测系统记录的所有“碎片 IP 数据包”报警有相同的源地址（Web 服务器地址）和源端口（Web 服务的端口）。报警的目的地址和目的端口是访问 Web 服务的客户端地址和端口。报警聚类将所有“碎片 IP 数据包”报警归类并产生一个“Web 服务器的 80 端口产生大量的碎片数据包”的泛化报警。这个泛化报警明确指出了产生报警的原因。但泛化报警并不总是能准确地指出报警产生的确切原因，它只为报警原因的分析提供帮助。报警原因分析常常涉及网络和应用环境的其它方面。在这个例子中，移除“碎片 IP 数据包”报警能大大减少报警数量，有利于在报警数据中发现其它更为隐蔽和有威胁性的恶意行为。

## 3　报警聚类

报警聚类把大量报警数据分为若干组，这样极大减少了报警数据量，简化了报警信息。通过为不同的报警聚类提出泛化报警，有利于分析报警原因。报警聚类方法直接影响了对聚类结果和报警原因的分析，但是目前并没有准确地实现报警聚类的方法，引起同一报警的报警原因常常是多方面的，并与网络和应用环境的实际情况相关。例如，一个有 TCP/IP 协议栈缺陷的主机，不断发出碎片化的 IP 数据包，假设这个主机位于路由器的后面，并且该路由器也可能将数据包分片。如果入侵检测系统在路由器前面实施检测，它会为碎片化的 IP 数据包发出报警，并且报警数据中的源地址表明碎片化的 IP 数据包来自有缺陷的主机。除非对网络系统有足够的了解，否则无法判断碎片化 IP 数据包的报警原因究竟是有缺陷的 TCP/IP 协议栈还是路由器分片。因此，仅仅给定报警数据并不能解决报警聚类问题。

### 3.1　近似报警聚类

由于很难实现完全准确的报警聚类，本文采用近似报警聚类方法。由于报警具有特定的结构化属性，如报警类型、报警类别、产生报警的数据包源地址、目的地址以及其它字段，在报警数据中寻找这些结构化属性就可以将类似的报警归类，从而实现近似的报警聚类。具有相同结构化属性的报警通常逻辑上由同样的恶意行为导致，因此具有相同的报警原因，并属于同一报警聚类。在网络入侵检测系统中，如果恶意行为具有某个明显的特征，则检测到该特征就认为发现了这个恶意行为。与这种方法类似，如果报警原因导致的报警具有某些特征，则具有这些特征的报警可能就是由这个报警原因导致的，归纳和抽象这些结构化属性就可以提出泛化报警。

下面通过例子说明结构化属性和泛化报警，泛化报警如表 1 所列。

表 1　4 种原因对应的泛化报警

| No. | Source IP | Source port | Destination IP | Destination port | Alarm type |
|---|---|---|---|---|---|
| 1 | Web server | 80 | Client | Non-privilege | IP Fragment |
| 2 | External network | Non-privilege | Web server | 80 | SYN Flood |
| 3 | External network | Non-privilege | Internal network | 80 | Red Code |
| 4 | FTP client | Non-privilege | FTP server | 21 | FTP SYST |

（1）安装有缺陷的 TCP/IP 协议栈的 Web 服务器产生碎片化的流量信息，该服务器响应客户请求时，会触发“碎片化 IP 数据包”报警；

（2）从外部网络对内部网络的 Web 服务器发起的分布式拒绝服务攻击触发“SYN 洪泛”攻击；

（3）红色代码（Red Code）感染的机器扫描内部网络，并探测存在安全漏洞的服务器，触发“红色代码扫描”报警；

（4）FTP 客户端对每个 FTP 连接发送 SYST 命令，触发大量的“FTP SYST 命令”报警。

第一行中，报警的源地址为 Web 服务器的 80 端口，目的地址为 Web 服务的客户端，端口为客户端的非特权端口，并且报警类型都为“碎片化 IP 数据包”，这些属性构成了报警内容，归纳的泛化报警指明该类报警的原因。第二行的报警来自外部网络的非特权端口，攻击 Web 服务器的 80 端口且报警类型为“SYN 洪泛”，这些属性归纳的泛化报警指明 Web 服务器受到 SYN 洪泛攻击。第三行的报警来自外部网络的非特权端口，指向内部网络的 80 端口且报警类型为“红色代码”，这些属性归纳的泛化报警指明内部网络受到红色代码的袭击。第四行的报警来自 FTP 客户端的非特权端口，指向 FTP 服务器的 21 端口且报警类型为“FTP SYST”，这些属性归纳的泛化报警指明 FTP 服务器受到 FTP SYST 命令的攻击。由此可见，报警聚类归纳的泛化报警能够表示出报警聚

类的主要特征。

为解决报警聚类问题，定义报警相异度 $d(.,.)$，报警的相异度描述了两条报警可以被归纳到同一泛化报警的合适程度。两条报警的相异度 $d$ 和它们的相似度成相反关系，如果两条报警 $a_1$ 和 $a_2$ 的相异度 $d(a_1,a_2)$ 越低，则它们越相似，越容易归纳为同一泛化报警。报警之间相异度的计算需要利用泛化关系图 $G$，图中属性的关系决定了属性之间的相异度。因此，首先考虑单个属性的相异度，然后通过各个属性的相异度计算报警之间的相异度。设报警属性为 $A_i$，对于 $x_1,x_2\in Dom(A_i)$，则 $x_1,x_2$ 之间的相异度 $d(x_1,x_2)$ 定义为在泛化关系图中通过公共的祖先 $p$ 连接 $x_1$ 和 $x_2$ 的最短路径，即：

$$d(x_1,x_2)=\min\{\delta(x_1,p)+\delta(x_2,p)\mid p\in G,x_1\nabla p,x_2\nabla p\} \quad (1)$$

其中，$\delta(.,.)$ 定义为在泛化关系图中两个节点之间的最短路径长度。例如，在图 1 中，$d(ip_1,ip_1)=0$，$d(ip_1,ip_4)=4$。然后通过各个属性的相异度定义报警之间的相异度。设报警 $a_1,a_2\in\times_{1\leqslant i\leqslant n}Dom(A_i)$，则 $a_1,a_2$ 的相异度 $d(a_1,a_2)$ 定义为属性相异度之和，即：

$$d(a_1,a_2)=\sum_{i=1}^{n}d(a_1[A_i],a_2[A_i]) \quad (2)$$

这个公式中没有考虑报警属性的权值，即某些属性可能在相异度计算中更加重要。相异度 $d(.,.)$ 衡量了报警被归纳为泛化报警的合适程度，也就是被归为同一报警聚类的合适程度。为了说明这一点，设 $g\in\times_{1\leqslant i\leqslant n}Dom(A_i)$ 是 $a_1,a_2$ 的泛化报警，则 $a_1,a_2\nabla g$，$d_i=(g,a_i)$ $(i=1,2)$ 表示在泛化关系图中把 $a_i$ 的属性移动到 $g$ 的对应属性时路径的总长度。如果 $d_1+d_2$ 的值越小，则用 $g$ 来归纳 $a_1$ 和 $a_2$ 就越适合；如果 $d_1+d_2$ 的值较大，则 $g$ 过于抽象，不足于捕获报警 $a_1$ 和 $a_2$ 的详细信息。因此可以用 $d_1+d_2$ 衡量其是否适合归纳为同一泛化报警。

除了定义报警之间的相异度 $d$ 外，还为报警聚类 $C$ 定义报警聚类的相异度 $H(C)$，当 $H(C)$ 较小时报警聚类可以归纳为泛化报警。设 $g$ 为报警聚类 $C$ 归纳的泛化报警，则在泛化关系图中 $g$ 是报警聚类 $C$ 中所有报警的父节点，即：$\forall a\in C$，$a\nabla g$。定义平均相异度 $d(g,C)$ 为 $g$ 和 $C$ 中所有报警的相异度 $d$ 的平均值，$H(C)$ 定义为：

$$d(g,C)=1/|C|\times\sum_{a\in C}d(g,a)$$

$$H(C)=\min\{d(g,C)\mid g\in X_{i=1}^{n}Dom(A_i),\forall a\in C,a\nabla g\} \quad (3)$$

平均相异度衡量了报警聚类 $C$ 归纳为泛化报警 $g$ 的合适程度，值越小，则报警聚类越适合归纳到泛化规则 $g$。当 $\forall a\in C,a\nabla g$ 且 $d(g,C)=H(C)$ 时，泛化报警 $g$ 覆盖了 $C$，这时 $g$ 最适合抽象 $C$。如果报警聚类中报警属性的所有泛化关系都为树形结构，则报警聚类存在覆盖。

在报警相异度和报警聚类相异度的基础上，形式化定义报警聚类问题为：设 $\Gamma$ 为报警集合，$\lambda\in N$ 是一个整数，$G_i$ $(i=1,2,\cdots,n)$ 表示每个报警属性 $A_i$ 对应的泛化关系图，则报警聚类问题 $(\Gamma,\lambda,G_1,\cdots,G_n)$ 是寻找集合 $C\subseteq\Gamma$，在 $|C|>\lambda$ 时使相异度 $H(C)$ 最小。这时 $C$ 就是一个报警聚类。也就是说，在所有满足 $C\subseteq\Gamma$ 的情况下寻找使得 $H(C)$ 最小的集合 $C$，如果同时存在多个 $C$ 满足条件，则任意一个都可以作为报警聚类。通过最小化 $H(C)$ 来保证报警聚类 $C$ 适合被归纳为泛化报警。参数 $\lambda$ 保证报警聚类 $C$ 具有适当的大小。在找到报警聚类 $C$ 后，就可以在 $\Gamma\backslash C$ 中继续寻找其它的报警聚类。

### 3.2 启发式报警聚类算法

本节构造启发式算法，算法寻找集合 $C\subseteq\Gamma$，满足 $|C|>\lambda$，但是 $H(C)$ 不一定最小。为简化问题，首先假定所有属性的泛化关系图都是树形结构。算法的伪码如下所示。

```
Function HeuristicAlarmCluster(Γ,λ,G)
  T=Γ//Store in table T
  for all alarms in T do  //Initialize count
    a[count]=1;
    while a∈T && a[count]<λ do
      Use heuristic to select an attribute Ai
      for all alarms a in T do //Generalize attribute Ai
        a[Ai]=father of a[Ai] in Gi
      while identical alarms a,a' exist do  //Merge identical alarms
        a[count]=a[count]+a'[count]
        remove a' from T
      end while
    end while
    output all a∈T with a[count]≥λ
  end function
```

算法首先将报警拷贝到表 $T$ 中，报警的每个属性 $A_i$ 在表中占一列，表 $T$ 还为每个报警记录一个用于计数的整型变量 *count*，并将所有报警的 *count* 初始化为 1，接着在循环中发现报警聚类。循环首先选择报警属性 $A_i$，把表 $T$ 中所有报警的属性 $A_i$ 替换为泛化关系图 $G_i$ 中它们父节点的值，这样原先不相同的报警可能变得相同；其次，如果报警 $a$ 和 $a'$ 的所有属性值都相等，则认为这两个报警相同，将两者的计数 *count* 相加赋给其中一个报警，并将另一个报警从表 $T$ 中删除，这样实现了两个报警的合并；循环直到某个报警的 *count* 至少为 $\lambda$ 为止。计数 *count* 始终代表了报警聚类中报警的数目。需要强调的是属性 $A_i$ 的选择，对每个属性 $A_i$，设 $F_i=\max\{f_i(v)\mid v\in Dom(A_i)\}$，其中，$f_i(v)$ = SELECT sum (*count*) FROM $T$ WHERE $A_i=v$，表示所有 $A_i$ 等于 $v$ 的报警的 *count* 之和。属性选择时选择 $F_i$ 最小的属性 $A_i$，也就是说选择了属性 $A_i$，则 $\forall j,F_i\leqslant F_j$。这种方法的基本思想是：如果报警 $a$ 满足计数大于 $\lambda$，则对所有的属性 $A_i$ $(i=1,\cdots n)$，有 $F_i\geqslant f_i(a[A_i])\geqslant\lambda$。

算法假定泛化关系都是树形结构，如果泛化关系是有向无环图而不是树形结构时，每个节点可能有多个父节点，这时在报警属性向父节点泛化时存在多种选择。这种多选择问题有两种基本的处理方法：一是用户定义规则确定唯一的选择，如在主机上同时运行 HTTP 服务和 FTP 服务时，HTTP 服务器地址和 FTP 服务器地址都可以作为主机 IP 地址的泛化，当目的端口为 80 时，IP 地址泛化为 HTTP 服务器地址，否则泛化为 FTP 服务器地址；另一种是并行处理所有可能的属性泛化，选择最先导致报警聚类达到 $\lambda$ 的泛化，但这种方法需要大量的计算，时间复杂度成指数数量级，因此在实际判断中很难采用。用户定义规则避免属性泛化的歧义，不仅可以降低计算的复杂度，而且考虑了网络和应用环境，因此报警聚类更加准确。

参数 $\lambda$ 的取值对算法结果有很大影响，选择的 $\lambda$ 过大，算法就会把不具有相同原因的报警归类到同一聚类中，这会导

致报警聚类过于庞大且难以解释，而且如果攻击者在攻击的过程中产生的报警数目小于参数 λ，则不会为这类攻击行为产生报警聚类；如果选择的 λ 过小，具有相同原因的报警就会被分割到不同的报警聚类中，从而增加报警聚类的数量和解释它们的时间复杂度。本文不对参数 λ 的选择做深入讨论，在实际应用中需要不断调节 λ 并检验报警聚类的结果，最后达到一个可以接受的值，在稳定的网络和应用环境中，λ 的值也是相对稳定的。

### 3.3 在 Snort 中的应用

在报警聚类方法应用于 Snort 的过程中，可以将 Snort 报警分类，实现对 Snort 报警的分析。Snort 报警属性的类型主要有 IP 地址、端口、数字、时间、字符串。在报警分析前为每种类型建立泛化关系。IP 地址的泛化关系如图 1 所示，针对网络布局和应用环境，将网络环境内的 IP 地址归类，建立层次关系，最终形成如图 1 所示的 IP 地址泛化关系图。端口的处理和 IP 地址类似，也要考虑端口的实际使用情况，建立端口泛化关系图。数字属性包括各种计数器（如“SYN 洪泛报警”中数据包的 SYN 序号）和大小域（如“大 ICMP 流量报警”中的数据包大小）等。通过建立一系列的数字范围并为数字范围命名，为数字属性建立了泛化关系，属于同一范围的数字都认为是等价的。例如，规定“小孩”、“青年”、“成人”、“老人”的年龄范围分别为[1,10]，[10,20]，[20,60]，[60,∞]，同时还规定“未成年人”和“成年人”的年龄范围为[1,20]，[20,∞]，因此“成年人”包含了“成人”和“老人”，这就为表示年龄的数字建立了分类并由泛化关系图表示。

人工定义泛化关系图的缺点是缺乏动态性，难以针对实际报警情况的变化而变化，而且对数字范围的划分难以把握，过于粗糙的划分和过于精细的划分都不利于准确地分析报警。时间属性主要指报警记录的时间，时间属性的泛化关系是为了区分如“周末”和“工作日”、“上班时间”和“休息时间”以及“月初”和“月末”这些有关应用的时间范围，如图 2 所示。

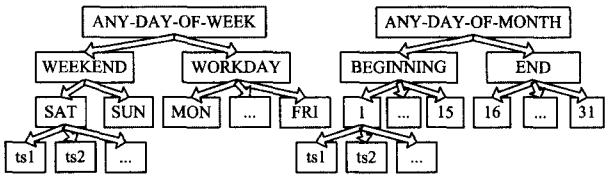


图 2 时间泛化关系图

有时需要把图 2 中的两部分合并为一个泛化关系图，这时一个时间节点就可能有两个父节点，一个为一周的第几天，另一个为一月的第几天。这就需要为时间的这种多个父节点问题建立在启发式报警聚类算法中选择父节点的规则。字符串是任意字符的组合，因此分析字符串属性的关键在于明确字符串值的语义。Snort 报警中的字符串包括报警的类型、类别和数据包内容。找出报警数据中可能出现的所有字符串，为字符串定义特征集合并建立从字符串到特征集合中特征的对应关系，这样报警中的字符串可以替换为特征集合的一个子集，从而建立字符串之间的泛化关系。如特征集{$f_1$，$f_2$，$f_3$}可以泛化为{$f_1$，$f_2$}，{$f_2$，$f_3$}或{$f_1$，$f_3$}，进一步又可以泛化为{$f_1$}，{$f_2$}或{$f_3$}，最上面的层次是空集，表示“任何特征”。在从报警数据的字符串中提取特征集合时也建立了它们之间的泛化关系，因此，这个泛化关系是动态建立的。

## 4 实验分析

实验中分析的报警数据包含 126,080 条报警。表 2 列出了 8 个最大的报警聚类对应的泛化报警。最后一列表示聚类中包含的报警数目。表中“任意”表示属性值被泛化到泛化关系图中的根节点，“未定义”表示在报警中没有明确记录这一项的信息，如 ICMP 协议中没有端口的概念，因此“ICMP 碎片攻击”报警中端口属性的值就是“未定义”，“ip1”，“ip2”等表示实际网络环境的主机。

表 2 8 个最大的报警聚类对应的泛化报警

| Alarm type | Source IP | Source port | Destination IP | Destination port | time | Data information | size |
|---|---|---|---|---|---|---|---|
| WWW IIS view source attack | External network | Non-privilege | ip4 | 80 | Any | Contain attack text | 54310 |
| WWW IIS view source attack | External network | Non-privilege | ip5 | 80 | Any | Contain attack text | 54000 |
| FTP SYST command attempt | External network | Non-privilege | HTTP/FTP | 21 | Any | Any | 4181 |
| IP Fragment attack | ip6 | Undefined | ip1 | Undefined | Workday | Undefined | 4581 |
| TCP SYN host sweep | ip1 | Non-privilege | Any | 80 | Any | Undefined | 761 |
| TCP SYN host sweep | Firewall | Non-privilege | Any | 25 | Any | Undefined | 253 |
| Fragmented ICMP traffic | External network | Undefined | ip4 | Undefined | Any | Undefined | 823 |
| Unknown protocol field in IP packet | ip7 | Undefined | Firewall | Undefined | Any | Undefined | 861 |

表 2 中列举的泛化报警包含了报警数据中 95% 的报警，简化了报警分析。但是报警聚类和泛化报警只是辅助报警原因的分析，还需要进一步寻找并确认报警原因。

对表 2 进一步分析，通过泛化报警查找报警原因。

表 2 中前两行表示的泛化报警的数据包信息中包含“GET /search_cgi/cgi? ction＝View&VdkVgwKey＝http%3A %2F%2Fwww%2Exyz%2Ecom”，这个非法的 HTTP 请求在报警数据中出现了一万多次。通过分析得出，当 HTTP 的 GET 请求包含“%2E”时就发生了“WWW IIS view source attack”攻击。报警原因在于 ip4 和 ip5 主机通过 Web 提供的搜索服务，它们返回到客户端的 URL 中都将点号替换为十六进制编码的%2E，当客户端点击这个 URL 时就触发了“WWW IIS view source attack”报警。

第三行表示的泛化报警为“FTP SYST command attempt”，强调很多 FTP 客户端发出 SYST 命令，在 FTP 协议中这个命令是合法的，用于返回 FTP 服务器的信息。这个泛化报警的报警原因在于客户端 FTP 软件的配置，使得 FTP 客户端每次发起连接请求时都发出 SYST 命令。

第四行泛化报警的原因在于 ip6 主机向防火墙恶意发送碎片化数据包来攻击防火墙，降低防火墙的处理能力。

最后一行是由于 ip7 主机使用了未知的传输层协议发送数据包，并试图让这些数据包通过防火墙。目前，有一些函数库（如 libnet 等）允许程序自定义数据包的内容，并将这些自定义的数据包发送到网络中，攻击者可以利用这些数据包建立秘密连接。分析表明，ip7 试图使用协议未知的数据包建立

（下转第 85 页）

数据查询的概率都必须满足以上两个特征。此外，通过比较实际的 eDonkey 网络，在模拟实验中，文件与虚拟服务器的数目约为 1∶200，虚拟服务器与物节点的比例约为 20∶1。

因为本文提出的算法适用于任何基于 DHT 的结构化 P2P 网络，为了实现的方便，我们在实验中使用 Chord 协议在 P2PSim 上建立了 P2P 网络，并分别运行 3 个负载平衡算法。对于 Chord 网络，通过在网络运行中的不同时间点，判断网络中超载的节点个数，取平均值，求得该网络的平均系统超载率。然后扩大 Chord 网络的规模，求得网络规模改变后 3 个负载平衡算法的效果，从而判断各种方法的可扩展性。经过模拟实验，得到图 6 的结果。

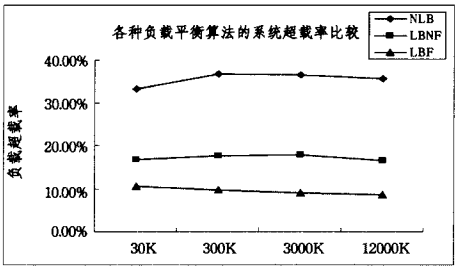


图 6　各种负载平衡算法的系统超载率比较

从图 6 可以看出，在不同规模网络运行中，分别使用这 3 种负载均衡算法，当网络规模变化时，系统超载率变化不大。说明这 3 种方法的可扩展性都很好。另外，我们提出的负载平衡算法，系统的超载率是最小的。说明采用基于存取频率的负载平衡算法，效果是三者中最好的。

**结束语**　本文针对结构化 P2P 系统提出了一个分布式的负载平衡算法。与现有算法相比，所提算法的主要改进有两点：①在路由表中加入了一个数据访问日志，用于保存历史访问记录并预测将来的访问频率。在选择被迁移的虚拟服务器时，考虑了被迁移的虚拟服务器的访问频率。②在选择迁移节点时采用完全分布式的方法，可以提高负载平衡算法的容错性，避免“单点失效”问题。

本文所设计的负载平衡算法是基于 DHT 的结构化 P2P 网络的增强机制。今后的工作主要集中在 P2P 网络的其他增强机制，比如拓扑一致性问题、安全问题以及无结构 P2P 网络的负载平衡问题。

## 参 考 文 献


[1] Rao A, Lakshminarayanan K, Surana S. Load Balancing in Structured P2P System[C]// Peer-to-Peer Systems II. Berlin: Springer, 2003, 2735: 68-79

[2] Xu Zhiyong, Bhuyan Laxmi. Effective Load Balancing in P 2 P System[C]//Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid. IEEE Computer Society Press, 2006: 81-88

[3] Fan L, Cao P, Almeida J, et al. Summary Cache a Scalable Wide-area Web Cache Sharing Protocol[J]. IEEE/ACM Transaction on Networking, 2000, 8(3): 281-293

[4] Chu J, Labonte K, Levine B N. Availability and Locality Measurements of Peer-to-Peer File System[C]// The International Society for Optical Engineering. Springer Berlin. 2002, 4868: 310-321

[5] Dabek F, Kasashoek M F, Karger D. Wide-area cooperative storage with CFS[J]. Operating Systems Review (ACM), 2001, 35 (5): 202-215

[6] Saroiu S, Gummadi P K, Gribble S D. A measurement study of peer-to-peer files sharing systems[C]// The International Society for Optical Engineering. Berlin: Springer, 2002, 4673: 156-170

[7] 李振宇，谢高岗. 基于 DHT 的 P2P 系统的负载均衡算法[J]. 计算机研究与发展，2006，43(9)：1579-1585

[8] 陈贵海，李振华. 对等网络：结构、应用与设计[M]. 北京：清华出版社，2007


(上接第 70 页)

秘密通信连接。

通过分析报警数据将报警聚类并归纳出泛化报警，进而根据网络和应用环境的实际状况分析出报警原因后，可以配置或增加安全组件来阻止类似恶意行为的再度发生。但并不是所有的报警原因都能得到有效处理，有的报警原因不完全在本地网络安全的控制范围内，或者修复报警原因指明的缺陷需要高昂的代价等。这些不能得到有效处理的报警原因仍将不断导致大量的报警，这些报警无疑加大了系统处理和分析的负荷，因此有必要根据报警原因对应的泛化报警构造报警数据过滤器，用于预处理报警数据，将那些属于不能有效处理的报警过滤，以提高分析效率。

**结束语**　由于恶意行为一般引起多次报警，且同一恶意行为可能多次发生，而这些报警有相同的报警原因，因此，本文把逻辑上相关的报警归类到同一个报警聚类中，进而归纳为泛化报警，并由它描述报警的共同特征，从而极大地减少了报警数量，有利于准确分析出网络和应用环境面临的安全威胁，以便针对这些报警原因采取应对措施。由于 $\lambda$ 参数对聚类结果有重要影响，因此，$\lambda$ 参数的选择还需要进一步的研究。

## 参 考 文 献


[1] Valdes A, Skinner K. Probabilistic alert correlation[A]// Proc. of the RAID[C]. Davis, CA, 2001: 54-68

[2] Cuppens F, Ortalo R. LAMBDA: a language to model a database for detection of attacks[A]// Proc. of the RAID[C]. Toulouse, 2000: 197-216

[3] Dain O, Cunningham R. Fusing a heterogeneous alert stream into scenarios[A]// Proc. of the ACM Workshop on Data Mining for Sec. Applications[C]. 2001: 1-13

[4] Ning P, Cui Y, Douglas R, et al. Techniques and tools for analyzing intrusion alerts[J]. ACM Trans. on Inf. and Syst. Sec., 2004, 7(2): 274-318

[5] Ning P, Cui Y, Reeves D. Analyzing intensive intrusion alert via correlation[A]// Proc. of the RAID[C]. Zurich, 2002: 74-94

[6] Ning P, Cui Y, Reeves D. Construction attack scenarios through correlation of intrusion alert[A]// Proc. of the ACM Conf. on Computer and Comm. Sec. [C]. Washington D. C, 2002: 245-254

[7] Templeton J, Levitt K. A requires / providers model for computer attacks[A]// Proc. of the New Sec. Paradigms Workshop[C]. Cork Ireland, 2000: 31-38

[8] Julisch K. Clustering intrusion detection alarms to support root cause analysis[J]. ACM Trans. on Inf. and Syst. Sec., 2003, 6 (4): 443-471

[9] Julisch K, Dacier M. Mining intrusion detection alarms for actionable knowledge[A]// Proc. of the ACM SIGKDD[C]. Edmonton, Alterba, 2002: 366-375