安徽大学

硕士学位论文

基于模式匹配的入侵检测系统研究

姓名：赵生艳

申请学位级别：硕士

专业：计算机应用技术

指导教师：仲红

2010-10

摘 要

随着网络的不断推广，网络安全问题越来越严重，网络安全检测技术也成为目前安全领域中研究的热点和重点。入侵检测（ID）技术是继防火墙、数据加密等传统安全保护措施后新一代的安全保障技术，在防护手段上更为主动，对计算机和网络资源上的恶意使用行为能够识别和响应。它在检测来自外部入侵行为的同时，也对内部的未授权活动进行监督。

目前，网络入侵检测系统（IDS）面临着诸多挑战，如何提高高速网络环境下的入侵检测系统的检测速度和如何降低入侵检测系统的漏报误报以提高检测准确性是其中最典型的两个问题。

本文分析了当前入侵检测技术存在检测效率低的问题，提出了提高效率的几种方法。由于目前大部分入侵检测系统都是基于模式匹配的，而模式匹配算法又是基于规则的，这种规则直接影响到系统的实时性和准确性。因此，本文对入侵检测中传统的单模式和多模式匹配算法进行了研究，提出了高效的改进的单模式匹配算法（改进的 BM 算法）和多模式匹配算法（改进的 AC_BM 算法）。针对改进的模式匹配算法，本文构建了相应的系统参考模型，该系统模型包括数据包捕获模块、数据包过滤模块、协议命令解析模块、模式匹配模块以及响应处理系统，并对各模块作了分析与设计。为了验证改进后的模式匹配算法的匹配效率，本文还用实验进行了说明。最后指明了未来入侵检测系统的研究方向。

关键词：网络安全；入侵检测；检测效率；模式匹配

Abstract

As networks have been popularized, the network security is more and more serious.Security of network technique in the detection technology also become a security area of research organizations and major. The detection technology is a new generation of security techniques and is even more active in protective measures. The detection technology can identify and respond to the computer and network resources is the malicious use behaviour. It is not only detected from the external invasion, but also supervise the internal unauthorized activities.

At present, network intrusions detection system (ids) is facing many challenges. How to improve the high speed internet environment of the detection system of inspection and how to lower the detection system failed to miss report are the two most typical questions.

This article analyzes detection inefficient about the current intrusions detection technology and proposes several ways of improving efficiency. At present,most of the intrusions detection system is based in pattern matching and Pattern matching algorithm is based on rule which has a direct impact on the timely and accurate. The text analysis the traditional pattern matching algorithm and comes up with Improvement of pattern matching algorithm.The text establishes a system of reference model based on Improvement of pattern matching algorithm which includes capturing data packet module, filtering data packet module, analysis agreement module, pattern matching module and Response system. The system analyes and designs all module. This article in the experiment verifies the improved efficiency of pattern matching and finally pointed out the future of the detection system for research direction.

Keywords: network security; intrusions detection; detection efficiency; pattern matching

第1章 绪论

随着计算机网络化的发展，信息全球化已经成为人类社会发展的大趋势。但由于计算机网络具有开放性、互连性，终端分布不均匀性和形式多样性等特征，使得网络信息安全日益成为一个至关重要的问题。现代化的建设离不开计算机网络系统，它为现代化的管理提供先进、快捷、可靠、安全的计算机网络环境。因此，如何确保计算机网络信息安全越来越成为人们比较关注的一个问题。

1.1 网络安全现状

网络的发展与普及，给人们带来了方便，但同时网络安全问题也时时威胁着人们，给人们带来了极大的困扰。网络安全目前现状[1,2,3,4]问题如下：

1. 局域网络速度快与负载重

目前局域网普遍采用百兆到桌面，千兆到主干互联。局域网集方方面面于一体，其规模大，且负载重，用户群少则数千，多则上万，而且这些节点一般安全防范措施比较脆弱，网络安全问题一旦爆发，则蔓延快，影响大。

2. 系统安全漏洞威胁

目前，网络环境中的软件系统或多或少地都会存在一些漏洞问题，加上恶意攻击与其他因素，不可避免地会遭到破坏。在不同的局域网中使用不同的数据库和数据库管理系统，其安全性也是不同的，加上软件开发商对软件安全性的要求也不尽相同，最终使软件的安全性也存在很大的差异。如果对软件安全性要求不高或考虑不够周全，必然给软件本身带来较大的安全隐患。

3. 计算机病毒肆虐

计算机病毒也给网络安全性带来了很大的威胁，当前是计算机病毒肆虐的时期，只要有计算机环境病毒就有可能存在。计算机病毒潜伏在计算机存储介质或程序里，在条件成熟的情况下，被激活的计算机病毒就可能对网络中的计算机资源进行一定的破坏作用。一般情况下，计算机病毒通过复制文件、传送文件、运行程序等操作进行病毒传播，我们日常的使用的可移动存储器和网络都是病毒传播的主要途经，尤其是现代普及的网络，加快了病毒的传播。目前我们见到的许多病毒基本上都是基于网络的，它为病毒的传播提供了便利，并且这些病毒破坏性极强，一旦遭到破坏，则可能造成系统运行瘫痪或崩溃，更严重的则可能使数据库中的数据遭到损毁，甚至数据丢失。比如2006年底出现的病毒“熊猫烧香”，

它严重地破坏了网络系统，给网络信息带来了很重的损失。又如 2007 年初出现的“金猪拜年”，是“熊猫烧香”病毒的变种，其进化速度相当快，对系统造成的破坏也是难以估计的。

4. 外部入侵

由于局域网络的开放式特点，使得局域网的管理也比较宽松，使得局域网容易遭受网络的入侵攻击。目前黑客入侵所带来的破坏性很严重，常见的是黑客将一些带病毒的文件在网络上进行传播，达到破坏对方数据的目的。黑客基本上都是想窃取对方你的机密数据，或者破坏其系统的正常运行，为了达到自己的目的，往往黑客会进行不断的信息轰炸，致使网络中的设备服务中断。Web 服务器或其他文件服务器也可能遭到攻击，一旦遭到攻击，其后果是严重的，不但使其数据达到删除或篡改的目的，甚至导致整个系统瘫痪甚至完全崩溃。可以说黑客的攻击杀伤力度较大、隐蔽性较强。

5. 内部攻击

据调查发现，80%的网络攻击都来自于内部，因为局域网络对内是开放的，内部人员对内部的计算机网络中各个部分是比较熟悉的，从而使得网络变得极为脆弱。内部人员通常对其所在的内部文化理解最为透彻，加上对内部网络基础设施及其支持系统的配置与操作的了解，这就使得由他们对网络实施攻击变得轻而易举。例如利用局域网内部开放的 IP 地址，内部人员能很容易地隐藏自己的 IP 地址，实施对内部网络的基本攻击，其攻击所带来的危害也是立竿见影的。其次，内部人员能够很容易地根据 IP 地址找到本单位网络所属的网段，这样就很容易运用 ARP 欺骗等攻击手段。目前，互联网上黑客攻击软件越来越多，攻击手法也越来越高明，越来越复杂，比如典型的 IP 碎片攻击、WinNuke 攻击等，这些基本上都是在获知 IP 地址的前提下实施对网络的攻击。

1.2 网络安全技术

为了确保计算机网络系统的安全，针对不同的情况采取不同的安全对策[3]，以防止来自不同方面的安全威胁。常用的网络安全技术措施主要有防火墙、杀毒软件、加密、认证授权、漏洞扫描、入侵检测等。

1. 防火墙

目前，保护局域网不受外界攻击的最有效的措施就是防火墙，防火墙是一种

逻辑隔离部件，它是建立在两个网络之间为了加强访问控制而设置的一个或一系列网络设备。通过防火墙可以进行封锁过滤，防止外部未授权的行为在被保护的内部网络中进出。由于防火墙是安置在网络边界的，它是计算机网络安全的第一道屏障，在确保网络正常的情况下，使用防火墙可以进一步强化内部网络的安全。防火墙是一种静态安全部件，它的访问控制是在制定好的安全政策下进行的，在高速网络环境下，攻击技术不断发展，攻击手段五花八门，显然这种技术已不能跟上目前或未来网络安全的步伐。防火墙是一种静态防御部件，不能主动跟踪入侵者，现在和将来要求防火墙技术必须发展与更新，单一的防火墙技术已不能满足需要，要求防火墙技术与其他安全技术联动，特别是与 IDS 的联动，也可以自身集成 IDS，实现一个全面的安全防御系统。

2. 数据加密

加密有公共密钥加密和对称密钥加密。数据加密是最核心的对策，是保障数据安全、可靠和完整性的最基本的技术措施和理论基础。数据在网络传输中尤其要进行数据加密，并对算法也进行保密，这是一种保障网络中的数据传输安全的最有效的方法。

3. 访问控制

访问控制主要是阻止未授权用户进入被保护的网络，这也是保障网络安全的的主要策略之一。通常情况下，访问控制有三种策略：强制访问控制策略、自主访问控制策略和基于角色的访问控制策略。网络权限控制技术、入网访问控制技术、网络服务器控制技术、属性控制技术、服务器控制技术、目录级控制技术以及防火墙控制技术等都属于访问控制技术。

4. 用户身份及身份鉴别技术

安全策略应包含如何识别用户。通常安全策略应规定用于用户 ID 的标准或定义标准的系统管理过程，安全策略还应确定对系统用户或系统管理员的基本的鉴别机制。如果机制是口令，则安全策略还应规定口令的最小字长、口令的最长和最短生存期以及口令内容的要求等。

在开发一个安全策略之前，每个组织应对管理员采用的机制做好决定，是否需要更强的机制。如果需要更强的机制，安全策略还应确定相应的安全要求。更强的机制对诸如拨号访问或 VPN 这些远程访问也是适用的。

5. 安全审计跟踪技术

可随时掌握网络中各工作站的工作状态，控制网络的运行，包括监视和记录系统的活动情况，为影响系统安全性的越权行为和存取留下线索，以便查出非法操作者；提供审计报告；检测和判断对系统的攻击，及时提供警报和防范；识别合法用户的误操作，并及时通知；随时掌握系统运行情况。

6. Virtual Private Network（VPN）技术

是将物理分布在不同地点的网络通过公用骨干网，尤其是 Internet 链接而成的逻辑上的虚拟子网。为了保障网络的信息安全，VPN 技术采用了鉴别、保密性、访问控制、完整性等措施，以防止信息被复制、泄露和篡改。

7. 漏洞扫描技术

漏洞即某个程序在设计时考虑不周全，当程序遇到一个貌似合理，但实际却无法处理的问题时，引发的不可预见的错误，漏洞威胁是目前网络安全的最大威胁，几乎网络中对安全的破坏都是由漏洞引起的。

为防止漏洞问题给计算机网络安全造成破坏，漏洞扫描是当务之急。漏洞扫描技术有 TCP/IP 扫描技术、UDP 扫描技术、ICMP 扫描技术等。目前常见的扫描技术是利用 TCP/IP 堆栈指纹的方法进行漏洞扫描的，该技术通过提取网络操作系统里的 TCP/IP 堆栈，以此堆栈作为“指纹”来确定系统的真正身份。这种技术的准确性较高，因为网络的堆栈参数位于系统的底层，很难被修改，目前利用这种技术实现的工具很多，有 NMAP、CHECKOS、QUESO 等。

8. 入侵检测技术

目前安全体系中引入了入侵检测技术[4,5,7]，它是一种动态的安全技术，对网络的易受攻击点和安全漏洞等能够积极主动地检测出来，在后果产生前提前探知入侵行为。IDS 检测能力极强，不仅能够检测出来自外部的入侵行为，同时也能够检测到内部用户的未授权活动。IDS 能够实现在网络节点上进行信息的收集，并完成信息的分析，根据分析出来的结果检测网络环境中是否存在一些违反安全策略的行为及遭到袭击的迹象。IDS 技术是一种积极主动的网络安全防护技术，它能够主动地防范网络中的恶意攻击行为及误操作，实时地提供了保护作用，将入侵拦截实施在网络系统受到危害之前进行，并进行实时的入侵响应。基于 IDS 的种种优点，人们对它的研究越来越重视。

1.3 课题研究目的及意义

随着办公自动化、数字化以及网络化的发展与普及，给用户带来前所未有的方便。但由于现代工作的业务管理模式、运行模式以及信息资源的服务对网络依赖程度的增大,加上它的共享性和开放性的不断发展，为信息管理带来了越来越多的不可预测的安全问题。如何确保网络化进程中网络设备、数据、系统的安全，促进基于网络的各项工作的正常运行，已经成为当今信息安全的重要课题。

目前各个领域采用的安全防护措施主要有：防火墙、数据加密、口令验证、VPN、代理服务器等，这些都是静态安全技术，如防火墙外围保护设备，只能对进出网络的数据进行分析，对网络内部发生的事件无法做出判断，而对入侵行为也只能进行人工实施和维护，不能主动跟踪入侵者。为了确保网络信息的安全，单纯的静态防御措施是不够的，必须采取更为主动有效的防护手段。目前的动态安全技术有入侵检测系统，是当前正在研究的热门安全技术，对网络的易受攻击点和网络上的安全漏洞问题能够进行积极主动地检测,对网络上的危险行为能够先于人工的探测。IDS不仅能检测出来自外部的入侵行为，对内部用户的未授权活动也能够检测到。IDS能够实现在网络节点上进行信息的收集，并完成信息的分析，根据分析出来的结果检测网络环境中是否存在一些违反安全策略的行为及遭到袭击的迹象。IDS是一种积极主动的安全防护技术，能够主动地实时保护网络上的恶意攻击及误操作，能够入侵拦截和入侵响应在网络系统受到危害之前。IDS集优点于一身，它的研究越来越受到人们的重视。近年来，IDS取得了很大的发展，但并不尽人意，主要的问题是入侵检测产品检测率较低，存在着很高的漏报率和误报率现象。漏报是指在入侵行为发生时IDS却没有发出报警，误报是指在入侵行为并没有发生时IDS报告发现了攻击。为了降低漏报误报现象，必须先分析一般入侵检测系统产生漏报和误报的原因，然后采取相应对策。

1.4 入侵检测技术的发展及应用

入侵检测系统研究的历史最早是1980 James Aderson的工作，在那时第一次提出了入侵检测这一概念，同时将入侵行为分为外部渗透、内部渗透和不法行为三种。为了检测用户对数据库的异常访问，1986年，W. T.Tener在IBM主机上利用COBOL完成的Discovery系统，是最早的基于主机的入侵检测系统的雏形。1987年，D.E.Denning提出了一个抽象的实时入侵检测系统模型，该模型就是著名的

IDES（Intrusion Detection Expert System，入侵检测专家系统）模型。IDS相对与传统的加密和访问控制安全策略而言，它是一套全新的计算机安全技术。这种技术下的模型由六个部分组成，分别是：对象、主体、活动规则、审计记录、异常记录以及轮廓特征。它完全独立于特定的系统平台、系统弱点、应用环境以及入侵类型，提供了一个通用的入侵检测系统框架。在1988年出现的Morris蠕虫病毒事件，激起了人们研究IDS的热潮。在1990年，加州大学戴维斯分校的L.T.Heberlein等人提出了一个基于网络的入侵检测——NSM（Network Security Monitor），它检查的不是主机系统的审计记录，而是通过局域网主动地跟踪网络信息流量来追踪可疑的行为。从此之后，入侵检测系统的研究开始了形成了两个重要的方向，一个是基于主机的IDS，另一个是基于网络的IDS。在1988年，莫里斯蠕虫事件发生之后，网络安全问题引起了学术界、企业和军方的高度重视。美国空军密码支持中心、劳伦斯利弗摩尔国家实验室、加州大学戴维斯分校、Haystack实验室在能源部、空军和国家安全局的共同资助下，积极展开了对分布式入侵检测系统（DIDS）的研究，并将基于主机和基于网络的入侵检测方法集成在一起实施。，DIDS与NADIR（Network Anomaly Detection and Intrusion Reporter）在1991年又提出了一种收集与合并处理来自多个主机的审计信息，以此来检测一系列主机的协同攻击。，Mark Crosbie和Gene Spafford于1994年提出了使用自治代理的方法来提高IDS的可维护性、可伸缩性、效率和容错性，这一方法非常符合正在进行的计算机科学其他领域（如软件代理，Software Agent）的研究。在1995年，开发了可以检测多个主机上的入侵的IDES完善后的版本——NIDES（Next-Generation Intrusion Detection System）。为了解决当时绝大多数入侵检测系统伸缩性不足问题，在1996年，又进行了GrIDS（Graph-based Intrusion Detection System）的系统设计和实现，该系统能够实现对跨过多个管理领域的大规模自动或协同的攻击进行检测。在2000年的2月份，对Yahoo、Amazon、CNN等大型网站的DDOS（分布式拒绝服务）攻击又一次引发了人们对IDS系统的新一轮的研究热潮。

1.5 本章小结

在本章中，主要对网络安全现状进行了分析，针对不同的安全问题所采取的不同的安全策略进行介绍；同时引入了目前网络安全策略中的新型技术：入侵检

测技术，就这一技术的研究目的和意义进行了说明；最后阐述了这一技术的发展历程及具体应用。

第2章 入侵检测系统概述

2.1 入侵检测系统定义

入侵检测系统[7]（Intrusion Detection System，简称IDS）是从计算机系统和网络系统中收集信息，通过收集的信息分析入侵特征的网络安全系统。IDS被认为是防火墙之后的第二道安全闸门，它与以往的安全措施不同的是能够先于造成危害之前检测到入侵攻击行为，一旦检测出攻击及时地利用报警装置与防护系统防范入侵攻击，尽可能地减少入侵攻击所造成的损失。若被入侵攻击，还可收集入侵攻击的相关信息，作为防范系统的知识，添加入策略集中，避免系统再次受到同类型的入侵，增强系统的防范能力。

2.2 入侵检测系统分类

入侵检测系统的分类[6,7,8,9]方法很多，基本上都是以“信息源、事件生成、事件处理、检测方法、入侵响应”等作为分类依据的。下面对入侵检测系统的分类作详细的介绍。

2.2.1 根据分析方法分类

根据分析方法分为误用检测和异常检测两类，原理是分别对其建立误用检测和异常检测模型模型。误用入侵检测是指利用已知系统和应用软件的弱点攻击模式来检测入侵，能直接检测到不利的或不可接受的行为。使用异常入侵检测技术来区分非正常的入侵行为的则是根据使用定量的方式描述可以接受的正常的行为特征来判断的。异常入侵检测能够根据使用计算机资源的情况和异常行为进行比较来检测入侵，它能够检测出与正常行为相违背的行为。

1. 误用检测（Misuse Detection）

误用检测是一种基于特征行为的类似于查毒的检测，首先建立一个相关的特征库，将收集到的非正常操作的行为特征与特征库中的特征比较看是否匹配，如果匹配，系统就认为这种行为是入侵行为，所以也称为特征检测。与特征库中相匹配的入侵都能检测到，但对于新的入侵行为或原入侵行为的变种不易检测到，极易产生漏报。

2. 异常检测（Anomaly Detection）

异常检测是一种基于统计分析原理的检测，根据正常的行为首先制定出正常

操作应该具有的特征（或叫用户轮廓），在这一个基础上试图用定量的方式加以描述以此建立正常活动主体的“活动简档”，将当前正在活动主体的活动状况与该“活动简档”相比较，如果主体的活动违反了该“活动简档”的统计规律，则认为该主体的活动很有可能是入侵行为。

异常检测不需要对每种入侵行为进行定义，其系统的效率完全取决于用户建立的轮廓的完备性和监控的频率，能有效检测未知的入侵，降低系统的漏报率。使用异常检测的系统能根据用户行为的变化进行自我调整和自我优化。但检测模型越精确，异常检测就会消耗越多的系统资源。一切违反其统计规律的用户活动都认为是入侵行为，极易产生误警。

2.2.2 根据数据来源分类

入侵检测系统需要以原始数据中包含的信息为基础来判断对其所监控的网络或主机的当前状态，入侵检测的分类方法很多，根据检测的数据来源的不同，入侵检测系统有主机型入侵检测系统HIDS、网络型入侵检测系统NIDS和混合型入侵检测系统。以下对三种类型的入侵检测系统进行介绍：

1. 主机型入侵检测系统

系统获取数据的依据是系统运行所在的主机。基于主机的入侵检测系统保护的是该系统运行所在的主机。HIDS通过对主机上的审计日志来监视与分析检测入侵，发生在该系统上的异常活动的证据会包含在这些日志文件中，这些被包含的证据指出了正有入侵或已成功入侵了该系统。在日志文件中能够发现入侵企图和成功的入侵，并快速地做出应急响应处理。

由于HIDS是运行在主机上的，占用的是主机上的资源，保护的是主机上的数据，对网络数据不敏感。但HIDS在主机上极易产生额外的负载，造成可移植性降低，因而使应用范围受到严重限制。

2. 网络型入侵检测系统

基于网络的入侵检测系统[10,11]主要是实时监控网络上关键节点的信息并采集信息，将采集来的原始网络包作为数据源实施数据分析。入侵检测系统能够实时监控并分析通过网络的所有信息，完成这一任务的是一个运行在混杂模式下的网络适配器，原始网络包的获得通过其他一些特殊硬件或软件也能实现。系统保护的是网络的正常运行，获取的是网络上传输的数据包。

利用基于网络的入侵检测系统可以提高侦测的速度，且隐蔽性好，占资源较少。但只能检测到本网段的活动，对于其他网段的活动不能检测，并且检测的精确度较低，在交换环境下难以实现配置，难定位入侵者，防入侵欺骗的能力较差。

3. 混合型入侵检测系统

系统主机型和网络型入侵检测系统都有各自的优缺点，基于主机的入侵检测系统能够更加精确地监视系统中的各种活动，使用基于网络的入侵检测系统则网络中的种种活动能够客观地反映出来，特别是系统审计的盲区都能够监视得到。基于主机和基于网络的入侵检测系统的结合便构成了混和入侵检测系统，两种技术结合能大幅度提升网络和系统面对攻击和错误使用时的抵抗力，使安全实施更加有效。

2.2.3 根据体系结构分类

1.集中式

集中式入侵检测系统有一个中央入侵检测服务器和多个分布于不同主机上的审计程序。通过审计程序可以把当地收集到的数据发送给中央入侵检测服务器进行分析处理。其可配置性和可伸缩性方面存在着致命的缺陷。随着网络规模的不断扩大，审计程序和入侵检测服务器之间传送的信息量就会越来越多，加大了负荷，导致网络性能大幅度下降。关键是一旦中央入侵检测服务器出现故障，整个网络系统则会陷入瘫痪。并且由于不同主机根据不同需求必须配置相应的服务器，配置过程也是非常复杂的。

2.等级式

该入侵检测系统将对划分的若干个区域分等级监控，检测系统与区域实现一对一的监控与数据分析，之后将分析的结果传送给上一级检测系统。但该系统的缺点就是一旦网络拓扑结构改变，必然导致区域分析结果的汇总机制的改变；并且，最终还要把各地收集到的分析结果传送到最高级的入侵检测服务器上进行全局分析，经过一级一级的传送，系统的安全性自然会降低。

3.协作式

该入侵检测系统将中央入侵检测服务器的任务分配给多个不分等级的基于主机的入侵检测系统，每个入侵检测系统负责监控本主机上的活动。提高了系统的安全性和可控性，但增加了所监控主机的工作负荷，且维护起来成本较高。

2.2.4 根据工作方式分类

根据不同的工作方式，入侵检测系统又分为在线检测和离线检测。

2.3 入侵检测技术

2.3.1 误用检测

误用检测[7,12]又称为滥用检测，其基本前提是假设所有的网络攻击行为和方法都具有一定的模式或特征，如果把以往发现的所有网络攻击行为的特征总结出来并建立一个入侵特征库，然后利用用户当前的行为和系统状态与特征库中特征进行模式匹配，如果匹配，则认为当前行为是入侵行为。

误用检测技术首先要制定出违背安全策略事件的特征库，判别所搜集到的数据特征是否在所搜集到的入侵模式库中出现，如出现则认为发现了违背安全策略的行为，这方法跟杀毒软件采用的特征码匹配原理类似。该过程的设计可以很简单，也可以很复杂。一般来讲，一种进攻模式可以用一个过程或一个输出来表示。它的优点是对已知的入侵行为检测的准确性较高，但对未知入侵模式的攻击行为的检测能力不足。

目前常用的方法具体有：专家系统误用检测系统、特征分析误用检测、状态转换分析误用检测、模型推理误用检测、条件概率误用检测、键盘监控误用检测等。

2.3.2 异常检测

异常检测技术又称为基于行为的检测技术，是入侵者活动异常于正常主体的活动，首先建立“正常”的行为特征轮廓，通过对当前用户的行为或系统与正常的行为特征轮廓比较是否有偏离来判断是否发生了攻击。它是一种间接的检测方法。

在建立用户的行为或系统特征轮廓的正常模型时，特征量的选取至关重要，既要能准确地体现用户的行为或系统特征，又要能使以最少的特征量涵盖用户的行为或系统特征。由于异常检测的参考基准是正常的特征轮廓，因此特征轮的廓参考阈值的选定是非常关键的，参考阈值是决定这一检测方法准确率的重要的因素，阈值设定的过小虚警率会提高，阈值设定得过大漏警率会很高，因此异常检测技术的难点就是如何设定“正常”的行为特征轮廓、选取特征量以及更新特征

轮廓等。这种情况下，极易造成异常检测虚警率的提高，但它可以有效地检测到未知的入侵行为。由于检测过程中需要实时地对用户或系统的特征轮廓不断地更新，需要的计算量就很大，因此要求系统的处理性能就会越高。

目前常用的异常检测方法有基于特征选择异常检测方法、统计异常检测方法、基于模式预测异常检测方法、基于数据采掘异常检测方法、基于神经网络异常检测方法、基于贝叶斯推理异常检测方法、基于贝叶斯网络异常检测方法、基于机器学习异常检测方法等。

2.4 入侵检测效率研究

2.4.1 产生漏报和误报原因分析

入侵检测系统(IDS)是一个多层安全体系架构的关键组成部分，其中漏报率和误报率是评价一个入侵检测系统（IDS）好坏的最常用的一个指标。入侵检测技术发展到目前为止其技术还不够完善，有大量的误报和漏报现象存在。漏报指的是攻击事件没有被入侵检测系统检测到，误报则指的是入侵检测系统将正常事件误认为攻击事件并对此行为产生报警处理。入侵检测系统中的漏报和误报问题不仅阻碍了 IDS 的进一步应用，也使得一些专家对 IDS 的存在价值提出怀疑。为了有效降低入侵检测系统的漏报率和误报率[13,14,15,16,17]，很有必要对入侵检测系统中的漏报和误报产生的原因进行详细的分析。

1. 漏报原因

入侵检测的发展经历了若干年，期间出现了基于免疫模型的入侵检测系统模型、利用神经网络构建分类器来提取特征和分类以及基于数据挖掘技术的入侵检测等方面探讨了入侵检测技术的实现问题。但采用的技术基本都是传统的模式匹配技术，为了提高入侵检测的检测性能，还必须具备大量或完备的审计数据集，增加了在匹配过程中的计算量大且训练时间较长等问题。模式匹配算法的匹配过程指的是将从网络上捕获的数据包中的第一个字节开始与系统的入侵特征库中的第一个特征串等长的一组字节开始比较。若正好完全匹配，则认为该数据包是一个攻击包，接着便产生报警；若内容不同或不完全相同，则从捕获数据包的下一个字节开始与没有匹配成功的入侵特征库中的串等长的串重新开始对比，如此重复，直到比较完位置。当每一组字节与某一特征串比较完后，数据包将后移一个字节后再与特征串重新进行比较，一直到数据包中的所有字节都对比完毕为

止。接着数据包与入侵特征库中的下一个特征串进行比较，重复上述匹配过程，直到匹配完入侵特征库中的所有特征串为止。只要特征库中有一个特征串与数据包匹配成功，则认为该数据包是一个攻击包。一个数据包匹配完之后，再从网络中读取下一个数据包进行同样的匹配操作。

网络中的每一个数据包都必须完成与入侵特征库中的每一个特征串的比较。最大的可能是每个数据包中的每个字节都要进行比较，此时运算量最大。如某个网络每秒钟流过1000000个数据包，且每个数据包按平均有100个字节进行计算，如果一个 IDS 其入侵特征库中有 10000 个入侵特征，按每个入侵特征串平均有 100 个字节计算，则根据以上的匹配过程，每秒钟需要匹配的次数为：$1000000\times100\times10000\times100=10^{14}$，可见计算量的巨大。

以上的数据是在最坏的情况下计算得到的，虽然可以通过一些改进的技术减少匹配次数，提高匹配速度，但是在高速网络环境下的宽带网络 NIDS 使用这种算法，其惊人的计算量是显而易见的。目前一般的计算机其运算能力都还不能达到这样的计算数值，因此很容易在检测过程中由于运算速度跟不上的原因而导致丢包现象产生，这样就产生了漏报。网络传输中的数据包越多越大，就越容易使入侵检测系统在检测时出现丢包，造成大量的漏报现象。最重要的是模式匹配采用的是与已知特征库中的特征进行匹配，对未知的攻击事件无法检测到，并且也不能检测出已知攻击的变种，因此可能导致漏报。

2. 误报原因

在匹配过程中由于匹配精度不高，极易产生误报现象的发生。如果在一次入侵中只是对入侵串做了一点小改动或有多条相同的入侵字符串，系统检测出第一条匹配时就会漏过剩下的入侵串，忽略了改动过的入侵字符串，因此产生会误报。由于一般的采用模式匹配的入侵检测系统检测攻击的方法是将捕获的数据包与入侵特征库中的特征串进行机械地匹配，无法判定攻击模式的真正目的和最终效果。也就是说简单模式匹配 IDS 在检测过程中，当一旦发现数据包中的内容与已知特征库中的攻击特征相匹配，此时系统就会发出警报，但许多被报警的活动行为最终是不会给系统或网络带来危害的，根本上算不上攻击，则相应的警报属于无效警报，入侵检测系统中正是由于这些无效警报的存在导致有着较高的误报率。

以上是误用检测技术产生的漏报和误报问题，异常检测技术也可能导致漏报和误报。使用异常检测首先要建立用户或系统的“正常行为”特征轮廓，因此在建立正常模型时，特征量的选取至关重要，既要能准确地体现用户的行为或系统特征，又要能使模型以最少的特征量就能涵盖用户的行为或系统特征。由于异常检测的参考基准是正常的特征轮廓，因此特征轮的廓参考阈值的选定是非常关键的。阈值设定的过小虚警率会提高，阈值设定得过大漏警率会很高，而统计方法中建立阈值的度是不容易把握的，参考阈值决定着这一检测方法准确率。因此异常检测技术的难点就是如何设定“正常”的行为特征轮廓、选取特征量以及更新特征轮廓等。极易提高异常检测的虚警率，但它可以有效地检测到未知的入侵行为。由于检测过程中需要实时地对用户或系统的特征轮廓不断地更新，需要的计算量就很大，因此要求系统的处理性能就会越高。

2.4.2 降低漏报和误报方法研究

1. 降低 IDS 漏报率方法研究

（1）分析模式匹配方法

目前，模式匹配[18,19]方法是很多入侵检测系统基本上都采用的方法，比如比较著名的 Snort[20,21]系统，模式匹配分析方法是入侵检测系统中常用的分析方法。

使用简单模式匹配方法的入侵检测系统消耗系统资源较多、检测速度慢且不准确，同时存在着以下严重问题：

①由于计算量大，容易造成数据丢包。关于计算量的计算前面已经叙述，这里不再重复。

②模式匹配检测技术使用的是具有固定的特征模式来探测攻击，只能实现唯一的和没有明确攻击特征的检测，会忽略那些做了变化的输入串。

③对那些看起来像不同的字符串的真实意图和最终效果，基于模式匹配的入侵检测系统不能够正确地作出判断。在一个基于模式匹配的入侵检测系统中，对于每一个字符串的要求是都必须在攻击特征数据库中增加一个特征记录。这样必然导致庞大的数据库产生，其实庞大的特征库实际上是没有必要的，只会徒劳地增加计算的负载，造成数据包丢包现象，一些存在于被丢的数据包中的攻击无法检测到，尤其是在当今高速网络环境下，漏报率会明显增大。

因此，在高速网络环境下的检测使用传统的单模式匹配方法已不能满足需求。为了适应高速网络环境下的入侵检测，数据包采用协议分析的技术也能够完成入侵检测，首先是对网络上的协议做了一个格式化、标准化、层次化的定义。通过网络协议的定义，再利用网络的层次性，在对网络数据包进行检测的同时，对网络协议一层一层地分析，对入侵检测效率的提高提供了很大的帮助。为了降低计算量，提高分析效率，最好的方法是在对数据分析时利用模式匹配和协议分析相结合，这样入侵检测的结果更加准确。这是目前常用的一种结合式的数据分析方法。

（2）分析协议分析方法[22,23]

网络的入侵检测系统目前最常见的入侵检测系统，检测的数据来源是网络上捕获的数据包，为了在不同主机之间进行相互沟通，要求网络数据包在网络传输过程中遵循预先约定好的协议，因此我们可以根据不同的协议对规则进行分类，产生不同的规则集。协议分析是根据已经存在的协议模式到原先规定好的位置上取值，而不像模式匹配一样进行逐字符地比较，接着根据取值情况对协议作进一步的判断，同时给出该协议实施下一步分析动作。协议分析方法能有效提高数据包的分析效率，同时对单纯模式匹配带来的误报问题还可以避免。

图 2.1 给出了协议分析方法的协议分类树。

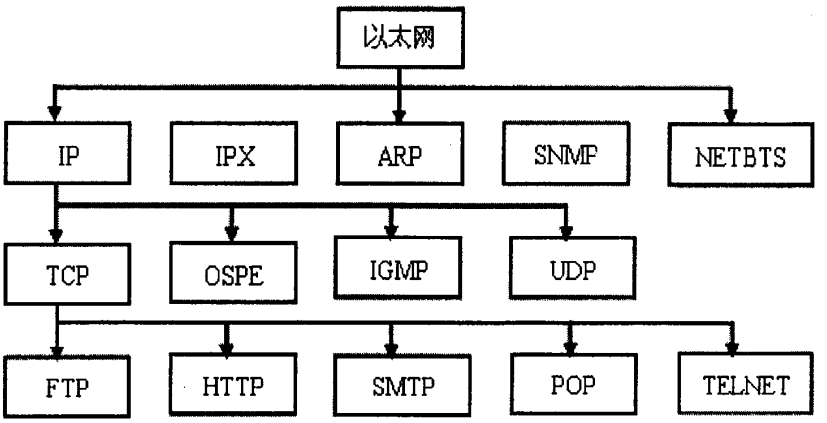

图 2. 1 协议分析树

在使用入侵检测系统进行模式匹配时，利用协议分析可以过滤掉许多不必要的规则，加快匹配的进程。数据包规则还可以根据不同端口实现第二次的分类，根据前面的两次分类情况，数据包能够快速地完成与特征库中的特征规则的比

较，大大降低了时间消耗。协议还可根据需要进行多次的分类，为了减小匹配的范围，分类时尽量地都在规则树上分叉。

数据包的分类是由协议分析机完成的，一般情况下协议名称、协议代号以及该协议对应的攻击检测函数共同组成了协议分析机的数据结构信息。协议的标志用协议名称表示，且是唯一的。为了加快分析速度用的编号用协议代号表示。为了细化分析数据，我们还可以在协议树中加入自定义的协议结点，以此来提高入侵检测系统检测的精确度，如可以把请求 URL 作为一个结点列入 HTTP 协议的树中，作为子节点的是 URL 中的方法。

协议分析机通过某一特定协议数据来完成分析，以此来查看分析的数据是否存在攻击。协议分析机一般情况下应该放在结构树的叶子节点上或尽量靠近叶子节，这样对提高检测的效率有很大的帮助，这样做的目的是尽量拉大协议分析机与根节点之间的距离，减少协议分析机的被调用的次数，因为协议分析机过多地被调用，整个系统的性能会受到影响。为了提高协议分析机的效率，需要对叶节点的协议类型进行详细的划分。

因此协议分析技术的采用使得检测速度和准确率都得到了提高，并且在检测过程中资源消耗也很少。

2. 降低 IDS 误报率[24,25,26]方法研究

（1）智能关联

为了降低检测系统的误报率，这里引入了智能关联的概念。所谓的智能关联是将网络 IDS 检测技术融入到企业系统的相关信息（如特征信息）中去。

目标主机上存在的与漏洞相关的几乎所有的告警信息都可以为使用智能关联的入侵检测系统提供参考。有主动和被动两种智能关联，通过扫描确定主机漏洞的是主动关联；借助操作系统的指纹识别技术来确定主机的漏洞是被动关联。目前，使用的主要是被动关联，被动关联被应用到指纹识别技术中，我们称其为被动指纹识别技术。以下给出了被动指纹识别技术的工作原理及工作流程。

①工作原理

匹配分析方法实质上也是被动指纹识别所采用的一种技术，其匹配双方一个是来自特征数据库中的目标主机信息，另一个是源主机数据流中的 TCP、IP 报头信息，比较两者之间的数据借以识别来自于源主机的数据是否存在恶意行为。

IP 报头信息一般由 DF(DontFragment)标志、数据报存活期（TTL）、窗口（WindowSize）以及数据报长（TotalLength）等。

DF 字段一般使用的是默认值。输入数据缓冲区大小决定着窗口的大小，由 OS 来 TCP 会话开始时设定。数据报在被丢弃之前所经过的跳数（Hop）被定义为数据报存活期，一个 TTL 值代表一个操作系统（OS），比如 TTL=12，OS=Windows；TTL=64，OS=UNIX。在 SYNACK 和 SYN 两种数据报中，OS 由数据报的长度（IP 报头与负载（PayLoad）长度之和）决定，比如数据报长度是 60，代表 Linux 操作系统；数据报长度是 48，代表 Windows2000 操作系统。如果 TTL 值是 64，初步判断 OS 可能是 UNIX 或者是 OpenBSD；再根据给定的窗口的大小（WSize）就可以区分到底是 Linux 操作系统还是 OpenBSD 操作系统。特征库中的一个特征信息可以用参数（TTL，WSize）来表示。

②工作流程

在入侵检测系统中使用指纹识别技术，可以降低系统的误报率，图 2.2 给出了它的工作流程。

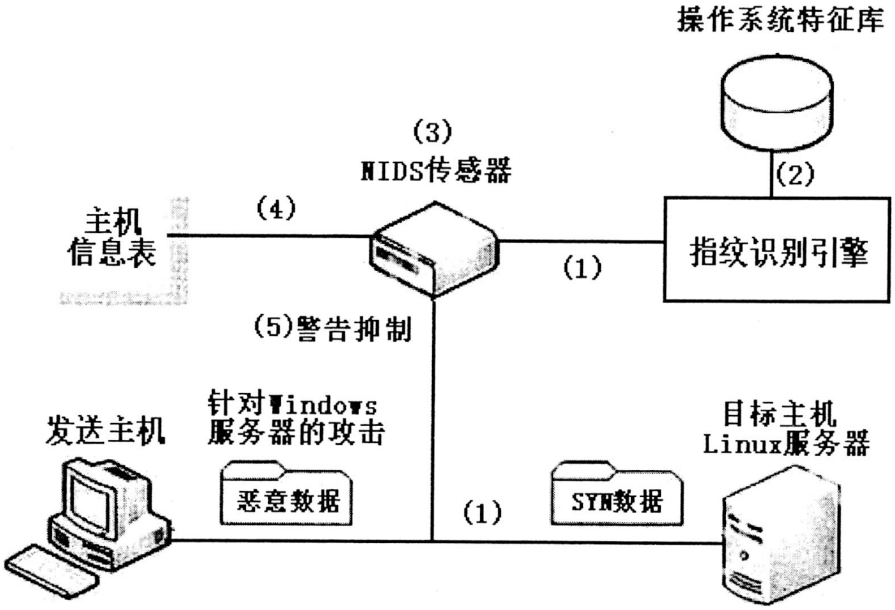

图 2.2 被动指纹识别技术工作流程

因此当某个数据包被 IDS 检测到具有一定的攻击信息时，判断目标主机上是否存在该攻击可利用的漏洞是通过查看主机的信息表知道的，IDS 还将抑制没有该攻击可利用的漏洞的告警的产生，与该漏洞有关的告警信息会被 I DS 记录下来，作为追究法律责任的依据。

（2）抑制告警泛滥

所谓“告警泛滥”是指短时间内产生的关于同一攻击的重复告警。告警泛滥抑制技术的使用能有效降低入侵检测系统的误报率。目前的入侵攻击大部分都是利用系统的漏洞，在短时间内入侵检测产品很容易对这些漏洞产生告警信息，一些告警信息却要被入侵检测系统的传感器重复记录。

网络上的某个流量短时间内由同一传感器产生的重复告警是通过 IDS 根据用户的需求减少或抑制。它首先在入侵检测系统的传感器中融入一些参数或规则，再由传感器识别并实施告警饱和的抵制操作。使用告警泛滥抑制技术，在告警前传感器可以对警报进行预处理。使用告警泛滥抑制技术的入侵检测系统不仅可以抑制告警，同时还将记录这些重复警告用于事后的统计分析。

（3）过滤警报

有效降低入侵检测系统的误报率一直以来成为了人们研究的热题。 很多时候，入侵检测系统的警报器发出的警报属于无效警报，这些警报中的入侵行为是不能给网络或系统造成破坏的，入侵检测系统的误报率正是因为这些不能造成破坏的入侵行为产生的无效警报。因此需要对无效警报进行过滤，如何过滤无效警报，前提是必须充分利用其所监控系统的各种环境信息，例如 UlfL indqvist 和 MagnusAlmgren 开发的一个应用型入侵检测系统，要求对网页联机进行检测，同时提取被检测的主机的信息，再加以检查，以减少误报。本文描述了使用新的警报过滤机制的网络型误用入侵检测系统。在该警报过滤机制中完成对安全信息扫描，同时建立一个网络漏洞数据库。一般的网络攻击基本上都是针对程序漏洞的攻击，只要程序的漏洞在网络中存在，则相应的攻击也就存在。在漏洞数据库中对网络中可疑行为进行查找，确认该漏洞是否存在，对于不存在漏洞的可疑行为只是记入日志而不发出警报，这样就可以实现无效警报的过滤。

①警报过滤机制体系结构

在警报过滤体系结构中引入了多个检测引擎，这些引擎主要布置在网络中的各个关键节点处，担负着网络数据包的捕获、特征匹配等工作，通过该检测引擎还可以将匹配不成功的信息发出警报，由警报过滤引擎对这些警报做进一步的过滤处理。警报过滤引擎对收到的警报数据首先写入到警报数据库中，然后查看漏洞数据库，看是否该警报的目标主机上是否存在警报所对应的漏洞，若漏洞存在

则发出警报，若漏洞不存在则警报无效只是将其记入日志中。另外为了获得网络中最新的网络漏洞信息，根据目前的安全状况安全扫描模块需要实时扫描，完成网络漏洞数据库的动态更新。

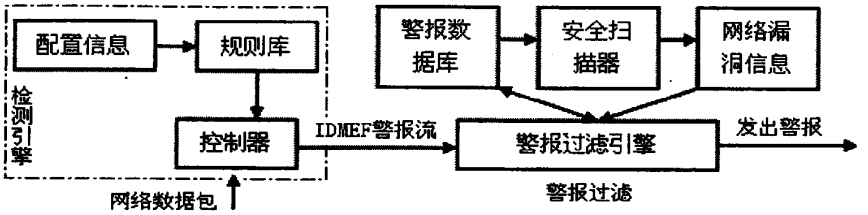

图2.3 警报过滤体系结构图

②IDMEF警报数据模型

报警的数据格式是五花八门，一个厂商生产的入侵检测产品对应着一个数据格式，可扩展性很差，同时增加了各安全设备之间的联动性。在这种情况下，人们提出了是不是应该对警报数据模型给出一个标准化的定义，于是一套基于Intrusion Detection Mes-sage Exchange Format (IDMEF)的标准化的数据模型诞生了，它是IETF委员会入侵检测工作小组（IDWG）于1999年给出的。在该数据模型中以面向对象的方式描述了一个警报所包含的所有的信息。

③网络安全扫描器

使用网络安全扫描器能够自动检测远程或本地主机安全漏洞的程序，远程或本地服务器的各种端口的分配情况和所提供的服务通过扫描都可以发现，还可以发现服务器上运行的软件版本以及软件存在的各种漏洞，为了解远程或本地服务器所存在的安全隐患提供可能。安全扫描器内部存在一个脚本漏洞检测脚本库，脚本与漏洞是一对一的关系，只须以该主机的地址为参数运行相应的脚本，就可以根据运行后的返回值知道某一台主机是否存在某一漏洞。

④警报过滤流程

首先将警报数据写入警报数据库，其次是确认警报的有效性，最后对漏洞数据库进行动态更新。在收到警报检测引擎发送来的警报数据时，警报过滤引擎首先要做的就是在警报数据库中将此警报数据录入该警报，这就是警报数据写入警报数据库的过程。警报数据写入之前先做好对冗余警报的过滤处理，这样做的主要原因是目前的入侵检测产品大部分都存在着警报冗余现象，这些警报冗余给入侵检测的响应系统准确地响应以及安全管理人员对警报的分析带来极大的困扰。

在本系统中对冗余警报消除的方法：一条冗余的警报被认为是在收到一条新的警报时，该警报的攻击类型、源地址、目的地址与警报数据库中的某一记录是相同的，并且时间间隔与警报已发生的数量在一定的范围之内。对该冗余警报的处理是将警报数据库中该记录已发生的数量加一，然后立即丢弃该警报，当该记录的数量超过了设定的阈值， 就将它交给警报过滤引擎，通过警报过滤引擎过滤掉此冗余的警报，同时该记录的数量值清零。若不是冗余的警报，在警报过滤引擎处理之前先将收到的警报写入警报数据库中。

一旦收到警报数据，警报过滤引擎是先将警报数据写入警报数据库中，再将与警报数据中记录本警报攻击所对应的漏洞然后提取出来，在漏洞数据库查找是否存在该信息，偌存在，被攻击的主机上就有这个漏洞，则发出警报，并提示该攻击存在的概率；否则直接将警报信息记入日志，该警报被认为是无效警报。为了确认攻击的有效性，通过安全扫描器对相应漏洞进行扫描，并将扫描结果写入漏洞数据库中，但这种情况无法确定被攻击主机上的漏洞是否存在。

为了更准确地检测到远程或本地主机的漏洞问题，对于漏洞数据库要进行动态地更新。漏洞数据库有若干张表组成，一台主机对应着一张表，每一台主机的漏洞信息、更新周期以及这个漏洞信息最近一次的更新时间都在每一张表上。图2.4 给出了漏洞数据库表的结构。

漏洞 CVE 号	漏洞状态	漏洞状态更新时间	漏洞状态更新周期

图 2.4 漏洞数据库

CVE 为已经暴露出来的弱点或者被广泛认同的信息安全漏洞给出一个公共的且唯一的名称，并给每个漏洞一个标准化的描述；漏洞状态的取值有三种：主机上无此漏洞，主机有此漏洞，不确定主机上是否有此漏洞；记录漏洞状态最近一次更新时间用漏洞状态更新时间表示；记录漏洞状态的有效期限用漏洞状态更新周期表示。如果记录中的“漏洞状态”属性值已过期，此时需要对漏洞数据库进行重新扫描并更新。主机的重要程度和漏洞的危险程度共同决定了漏洞状态更新周期，更新的周期根据主机上的服务的重要性或漏洞的危险性决定的，服务越重要、漏洞越危险则状态更新周期就越短。起始的漏洞数据库是通过一次完全的

安全信息扫描，随着网络中漏洞信息的变化，该漏洞数据库中的数据也会发生变化，可能会使现在的数据库与原先数据库数据不一致的情况。

文中漏洞数据库的动态更新其更新策略分为两个步骤：第一步骤当一个警报被传送过来，且满足一定的条件时，向安全扫描器提交扫描请求，并将扫描结果写回数据库中，起到更新数据库的作用，同时更新数据库记录的更新时间。如果只是局部更新网络当前所受到的攻击信息的漏洞数据库，很可能对数据库中其它漏洞状态不能及时更新，如果让攻击者转了空子，照样可以实施入侵。但如果对漏洞数据库进行全面的更新，又会增加系统的负担，因为这种全面的更新涉及到的主机和漏洞数量都比较大。第二步骤的更新实际上是建立一个具有扫描单元组的扫描更新队列，其中的每个扫描单元组中都有两个列表组成的，一个是待扫描的主机，一个是待扫描的漏洞。扫描队列每次选出具有最高优先级的单元组进行扫描，它的调度形式与操作系统中的进程调度很类似，同时完成漏洞数据库的更新。不同的网络对应的单元组也不相同，根据单元组所对应的服务的重要性和所受到攻击的频繁程度给每一个单元组确定一个优先级。

（4）告警融合

利用告警融合对解决漏报误报问题是及其有利的，它是将具有相关性的由不同传感器产生的低级别告警融合成高级别的告警信息称为“告警融合”。低级别的告警可以促使高级别的告警产生，前提是当低级别的告警满足一定的条件。当单独考虑每一次的扫描时，则扫描出来的事件都可能认为是独立的事件，可以忽略不计对整个系统造成的影响。在短时间内，如果把扫描产生的一系列事件综合考虑，会有不同的结果。通过扫描器扫描，如果入侵检测系统在短时间内检测到来自于同一 IP 的事件，甚至不断升级，则该事件可能是攻击前的渗透操作，应该引起网络管理员的重视，将当前事件作为高级别告警对待，使用告警融合技术可以发出早期攻击警告。在没有告警融合技术的前提下，网络管理员可以根据实际情况判断低级别的告警是否是高级别攻击的前兆。元告警规则中定义了端口号、事件类型 IP 地址、事件数量、事件顺序、时间窗等参数，通过对元告警相关性规则的设置，网络管理员可以完成高级别的警告处理。

2.5 入侵检测系统模型

CIDF(Common Intrusion Detection Framework)将入侵检测系统需要分析的

数据统称为事件（Event），事件可以是系统日志里的数据，也可以是网络中的数据包或其他途径得到的数据。所谓入侵检测其实就是检测系统和计算机网络以发现违反安全策略的过程。入侵检测通过网络数据包或信息的收集，检测可能的入侵行为，并且在入侵行为构成危害之前能及时地发出警报通知安全管理员并进行相关的处理措施。为了实现这一目的，入侵检测系统应具备 3 个必要的功能组件：数据源、分析引擎以及响应组件。事件产生器、事件数据库、事件分析器事件响应单元共同构成了入侵检测系统[7,9,27,28]。所谓的事件既可以是日志文件中的信息，也可以是网络上的数据包。关于以上四个组建现作说明如下：

①事件产生器：从整个计算机环境中获得事件，并向其他部分提供该事件。

②事件分析器：对事件产生器获得的数据进行分析，并产生分析结果。

③事件数据库：存放各种中间和最终数据的地方，即可以是简单的文本文件也可以是复杂的数据库。

④响应单元：对分析结果做出反应的单元，它可以做出切断连接、改变文件属性等反应，也可以进行简单的报警。

其中的事件数据库用数据库的形式表示比较方便，事件的产生器和分析器以及响应处理单元都可以以程序文件的形式表示。图 2.5 给出了入侵检测系统通用模型。

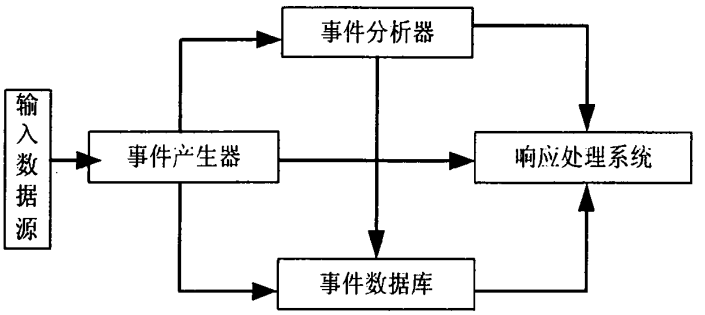

图 2.5 入侵检测系统的通用模型

1．事件产生器（Event Generators）

入侵检测最基本的是收集信息，这些信息可以来自于整个计算机系统，由事件产生器完成。入侵检测系统检测效果的好坏在很大程度上依赖于信息收集的可靠性、正确性和完备性，因此必须确保采集和报告信息的工具的可靠性。主机型入侵检测系统的数据源主要是主机的审计日志文件。网络型入侵检测系统的数据

源主要是网络中关键节点处的信息，通过实时监控侦听网络上的数据包来采集数据，进行攻击分析的数据源是原始的网络数据包。实时监视和分析所有通过网络进行传输的这些信息可以通过网络适配器来完成，一旦检测到入侵行为，由响应模块对攻击做出应有的反应，主要是负责通知、报警甚至中断连接等方式。基于网络的入侵检测系统一般安装于网络边缘的关键节点处。

2．事件分析器（Event analyzers）

接收来自于事件产生器的事件信息，对其进行分析，判断事件是否为异常现象或入侵行为，最后对判断结果转变为告警信息。事件分析器是入侵检测系统的核心组件，它的好坏直接影响着入侵检测系统的性能，事件分析器的分析方法有如下三种：

①模式匹配：将从主机的日志文件或网络上收集到的信息跟入侵特征库中的特征比较，以此来发现有没有违背安全策略的行为。

②统计分析：首先给系统对象创建一个统计描述，统计正常使用时的一些测量属性（如操作失败次数、访问次数和延时等）；测量属性的平均值和偏差将被用来与系统、网络的行为进行比较，其值如果超出正常值范围，就认为有入侵发生。

③完整性分析（主要用于事后分析）：主要关注某个文件或对象是否被更改。

3．事件数据库（Event databases）

事件数据库是存放各种中间和最终数据的地方，即可以是简单的文本文件，也可以是复杂的数据库。庞大而复杂的数据一般都采用成熟的数据库产品来支撑。其他系统模块可以对事件数据库中的数据进行添加、删除、访问、排序和分类等操作。

4．响应单元（Response units）

响应单元主要是当事件分析器发现有入侵迹象时及时做出相应处理，这种处理可以是作出切断连接、改变文件属性等强烈反应，也可以只是简单的报警。而响应的对象并不局限于可疑的攻击者。

2.6 入侵检测系统存在的问题及发展趋势

2.6.1 入侵检测系统存在的问题

入侵检测系统是安全防御的第二道屏障，在安全方面起着重要的作用，但我

国用户对它的认知程度还不够，加上入侵检测技术是一门全新的技术，很多方面还等待我们去研究和开发，研究和开发入侵检测产品不是每一个厂商都能够做得到的。以下从几个方面讨论目前入侵检测产品普遍存在的问题[1,2,3]：

1．漏报和误报问题

目前主要的入侵检测产品都是基于网络的，其数据源主要是网络数据包，入侵检测系统首先从网络上截获数据包，并进行分析，如果检测出来的攻击行为是一次对系统进行的攻击尝试，这代表了一种攻击的企图尝试攻击是不是需要报警，得有系统管理人员作出准确地判断。但大量的报警会致使安全管理员精力分散，对于真正的攻击反而无法作出反映，极易产生漏报。随着网络入侵者水平的逐渐提高，不断更新的攻击手法，入侵检测系统能否准确地报出所有的入侵攻击，这是我们目前和将来要研究的。

2．隐私和安全存在矛盾

为了检测网络中的数据包，必须度网络上的所有截获的数据进行分析，这一点对防范入侵极其重要，但用户的隐私可能会存在一定的风险。到底对网络中什么样的数据进行检测，就要看具体的入侵检测产品是否能提供相应功能以供管理员进行取舍。

3．主动发现与被动分析存在矛盾

入侵检测系统对网络中存在的安全隐患无法主动发现，只能采取被动监听的方式发现网络入侵的。如何主动发现入侵攻击也是入侵检测产品面临的重要问题。

4．处理上的速度问题

入侵检测系统首先是截获网络中每一个数据包，并对其进行分析、匹配以查看是否具有某种攻击的特征，这一过程需要花费大量的时间和系统资源。这样必然对处理速度有一个更高的要求，否则影响入侵检测系统检测的准确性和时效性。当前 IDS 产品大都时百兆、千兆的，其性能指标与实际要求还存在很大差距。随着核心网络带宽容量和用户对带宽要求的不断增加，高速网络环境下的入侵检测系统对数据包捕获和分析的能力都会明显下降。如何提高入侵检测速度以适应高带宽、高流量的网络已经成为亟待解决的问题。

5．功能性和管理性存在矛盾

随着入侵检测技术的发展，入侵检测产品的功能也在不断地增加，在增加功能的前提下能否不增加管理员的管理难度也是一个急需要研究的问题。入侵检测系统是否容易部署；入侵检测数据库能否在不需管理员干预的前提下对所有数据进行自动备份和维护；入侵检测系统采用何种报警方式；入侵检测系统自身安全性如何等，这些都需要我们积极地去考虑。

6．单一性的产品与复杂的网络应用存在矛盾

开发入侵检测产品最初的目的是为了检测主机或网络上入侵攻击，根据目前和未来复杂的网络应用需求，仅仅检测这些入侵攻击是远远不够的。当网络出现问题时，网络管理员很难分清到底如入侵攻击引起的还是网路自身的问题，如果是入侵行为，对入侵行为作出什么样的处理，以及入侵检测产品能否与目前网络中的其他安全产品进行联动等，这些都是我们急需要解决的问题。

2.6.2 入侵检测系统的发展趋势

随着网络技术和网络规模的不断发展，人们对计算机网络的依赖程度也在不断增强。但同时针对网络系统的攻击行为也越来越猖獗，攻击者的水平越来越高，攻击手法越来越复杂。入侵检测系统会随着网络技术和相关学科的发展而变得日趋成熟，其未来发展的趋势[1,2,3]主要表现在以下几个方面。

1．分析技术的改进

目前的入侵检测系统或多或少地都存在着漏报和误报现象，这一现象要解决必须对分析技术进行改进。目前常见的入侵检测分析方法包括协议分析、模式匹配、行为分析、统计分析、数据重组等。

模式匹配是目前入侵检测系统中用得最多和最广泛技术，不论是在商业入侵检测系统中还是在其他入侵检测系统中，模式匹配算法都以其准确和实用的优点成为入侵检测系统中使用的主要技术。为了解决单模式匹配算法在处理速度上存在的严重缺憾，模式匹配算法有待进一步改进。协议分析技术是将具有严格定义格式的数据包按照各层次协议报文封装的反向顺序，层层解析出来。主要是查看网络数据包中各层协议字段值是否在网络协议定义的期望值符合里，若在期望值里，则系统认为当前数据包具有攻击的可能性。但纯粹的协议分析对分散在不同数据包中的攻击事件无法检测到，必须对网络数据流进行重组，然后重新分析重组后的数据，利用此技术能有效地降低误报和漏报。如果待分析的行为比较简单，

采用行为分析技术比较合适。行为分析技术在目前还不够成熟，但行为分析技术能够根据前后发生的事件分析是否有攻击存在和攻击能否生效，这也是我们为什么要研究它的原因，行为分析技术为未来分析技术的发展提供了一个方向。统计分析是异常入侵检测中使用最广泛的且较为成熟的技术。统计分析首先通过异常检测器观察主体的活动，根据主体的活动情况用定量的语言描述出这些主体的正常行为轮廓，将主体正在的行为轮廓与已储存的正常轮廓进行比较，判断其异常行为。但在异常检测中正常行为轮廓建立的尺度不好把握，如何找到一个合适的轮廓阈值有待进一步研究。入侵检测的准确程度与规则库是否能及时更新也有很大的关系。为了充分发挥入侵检测系统的优势，目前最好的办法是多种分析技术的融合。

2．网络审计和内容恢复功能的引入

最高境界的入侵检测是基于行为分析技术的检测，但目前行为分析还不够成熟。使用内容恢复和网络审计的功能也可以完善入侵检测。内容恢复是以协议分析为基础的，它可以使网络中发生重组的数据加以完整恢复和重新记录，内容恢复技术的使用可以完整地监视到网络上发生的一切可疑行为。网络审计则是记录网络中所有的连接事件。入侵检测系统中的网络审计是由入侵检测的接入方式决定的，同时还可以记录网络中进出的信息和网络内部的连接状况，这一点类似防火墙，对内容恢复而又无法恢复的加密连接使用网络审计功能尤其有用。内容恢复和网络审计的使用主要是调动管理员参与行为分析，能够切实让管理员看到网络的真正运行状况。也就是管理员在看到攻击事件报警的同时，还能看到整个攻击的过程，对攻击造成的危害有个一定的了解。但使用此功能时应该注意保护用户的隐私。保护网络审计和内容恢复的使用，既可以发现外部的攻击，也可以发现内部的恶意行为，既可以发现已知攻击，同时对未知攻击也能探测。

3．安全性能和易用性能的提高

一个安全产品的好坏首先是看自身的安全问题，如果自身的安全都得不到保障，开发出来的这个产品就失去了意义，目前，大部分入侵检测产品都免除了自身的安全问题，采用的都是黑洞式接入、硬件结构的方式。一个产品的好与否，易用性也是一个指标，随着入侵检测技术的发展，人们对入侵检测产品的易用性

要求也在不断提高，如数据库的自动维护、多样的报表输出和全中文的图形界面等。优秀入侵产品就应该具有这些特性，这也是以后继续发展细化的趋势。

4．集成网络分析和管理功能的引入

使用入侵检测系统不仅能检测出网络上存在的攻击，同时网络中所有的数据还可以被接受得到，在分析网络故障和管理健康方面也可起着很重要的作用。某台主机一旦出现问题，能及时地对该机器进行管理。入侵检测采用的分析方法有被动分析和主动分析两种。今后安全方面的发展方向就是入侵检测产品集成扫描器（Scanner）、嗅探器（Sniffe）和网管等功能于一体。

5．巨量数据处理技术的改进

在高速网络环境下产生巨量的数据，要求入侵检测系统的性能也必须不断地提高，要求检测速度进一步的提高，目前百兆、千兆入侵检测产品的出现则顺应了这一要求。为了完成在千兆环境下的检测工作，入侵检测产品必须具备分析攻击的功能和网络审计与内容恢复的功能。目前，比较通用的做法是将网络数据分流，这样可以减轻巨量数据的处理负担。

6．防火墙联动技术的启动

目前，防火墙联动技术还只是一个理念，但如果能对这种技术加以很好地利用，则能够将检测出来的入侵攻击自动地发送给放火墙，然后防火墙启动动态规则，对发送给防火墙的入侵行为进行及时地入侵拦截。但目前这种技术还没有得到完全实用的阶段，随意的使用会导致更多问题的出现。无限制的使用联动，对防火墙的稳定性和网络应用也会造成负面影响。目前防火墙联动功能只在某种场合下起作用，但随着技术的不断提高提高，联动功能将会日益趋向实用化。

7．入侵检测系统的标准化

最近几年入侵检测产品的市场得到了快速的发展，越来越多的公司相继地对这一领域展开了研究。到目前为止，入侵检测系统还没有一统一的标准，不同厂家生产的IDS产品之间的数据交换和信息通信几乎不可能，难以达成联动。针对这一情况，美国成立了一个入侵检测工作组（IDWG）对IDS规范化提出了一系列标准草案。但也只是制定入侵检测响应系统之间共享信息的数据格式和交换信息的方式。。国防高级研究计划署（DARPA）提出了通用入侵检测框架（CIDF），介绍一种通用入侵说明语言（CISL），用其来表示系统事件、事件分析结果和响

应措施。为了把 IDS 从逻辑上作为一种面向任务的组件，CIDF 试图规范一种通用的语言格式和编码方式来表示在组件边界所传递的数据。CIDF 有四个部分组成：IDS 的体系结构、描述语言、通信体制和应用编程接口（API）。目前，IDWG 的规范标准和 CIDF 都还不够成熟，仍在处在不断地改进和完善中，但标准化是入侵检测系统发展的必然方向。

2.7 本章小结

在本章中，给出了入侵检测系统的概念；对入侵检测系统进行了分类；并详细介绍了入侵检测系统中的重要的两种技术：误用检测技术和异常检测技术；给出了入侵检测系统的 CIDF(Common Intrusion Detection Framework)模型；就漏报和误报问题进行了分析，针对具体原因给出改善漏报和误报的方案；最后就这一系统目前所存在的问题进行剖析，并就当前和未来的网络安全问题提出了入侵检测系统的未来发展的趋势。

第 3 章 模式匹配算法

所谓的模式匹配[21,23]，就是将通过不同渠道收集到的信息与模式数据库进行比较，该过程可以很简单，也可以很复杂，但事先必须建立相应的检测规则，比如指定 IP 地址、端口号、标志位等，类似于防火墙，它与防火墙不同的是入侵检测系统的规则特征还可以指定网络数据包中是否包含特定的特征字符串，且其检测引擎的计算量远远大于防火墙。目前，大部分入侵检测系统基本上都是基于规则的入侵检测系统，比如当前比较流行的 Snort 和 NFR（Network Flight Recorder）系统，而在入侵检测领域中最广泛使用的检测手段和机制之一就是模式匹配。

3.1 单模式匹配算法

3.1.1BM 算法

1. 算法原理

BM 算法是一种快速单模式精确匹配算法[]，其基本思想是从右向左逐字符进行比较。首先将文本串 T 与模式串 P 左端对齐，从模式串的右端开始向左逐个字符进行比较，一旦某趟比较不匹配，BM 算法就会采用两个启发性函数：坏字符启发函数 Badchar（）和好后缀启发函数 Goodsuffix（），根据这两个函数计算出模式串右移的距离（偏移量），以最大的距离右移文本串中的指针，直到匹配成功为止。因为在实际应用时，正文中大部分字符根本不出现在模式串中，即几乎所有位置都匹配的情况很少，所以应用 B M算法可以大大加快匹配速度。假设有一长度为 d1 的输入串 T，长度为 d2 的模式串 P，匹配的目的是确定 P 在 T 内的位置，或者 P 不在 T 内。假如 T 和 P 字符的编号都是从 0 由左至右进行的，要说明 P 在 T 的 d 处匹配，必须其中的每个字符 Pi 对一个指定的偏移 d 都匹配相应的字符 Ti+d。

（1）坏字符移动

若考虑 Badchar 函数移动的约束，为了使文本串 T[i+j]和它在模式串 P[i]左侧从右至左首次出现的位置对齐，模式串必须右移，这时 P[i+1...m-1]＝T[j+i+1...+m-1]＝m 且 P[i]≠T[j+i]时。图 3.1 给出了对齐方式。

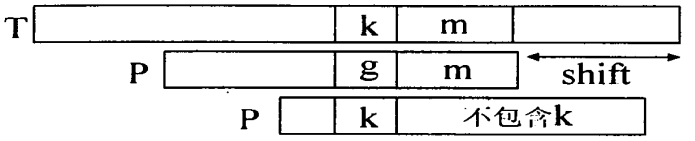

图 3.1 k 出现在模式 P 中的坏字符移动

如果模式串 P 中没有出现，要右移模式串，使得模式 P 最左端字符与 T[j+i+1] 对齐，必须满足文本串中的 T[i+j]在 P 中找不到，结果如图 3.2 所示。

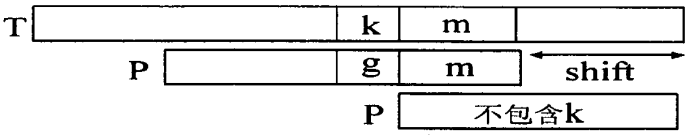

图 3.2 k 不出现在模式 P 中的坏字符移动

（2）好后缀移动

若考虑 Goodsuffix 函数移动的约束，首先在模式串 P 中查找在 m 左侧且与该后缀相同的片段（P[i+1...t-1]=T[i+j+1...+t-1]）之后，再将后缀与该片段对齐。当字符出现不匹配时 P[i+1...t-1]=T[j+i+1...+t-1]=m 且 P[i]≠T[j+i]，这时假设模式串中的 P[i]=g，文本串中的 T[i+j]=k。图 3.3 给出了这一过程。

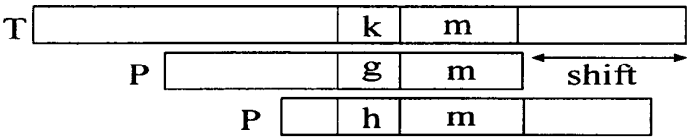

图 3.3 片段 m 重新出现的好后缀移动

m 为一个已经匹配的后缀，如果模式串 P 中不存在只有一个后缀 m 的片段，则利用这一后缀右移模式串，尽可能的使在之后出现的一个既是已匹配的后缀又是模式前缀的字符串 n 的值变大，图 3.4 给出了这一过程。

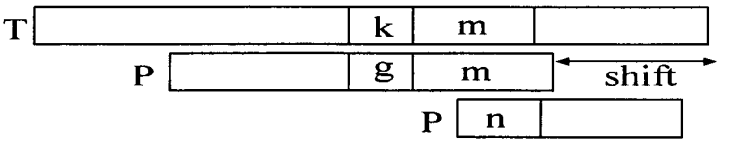

图 3.4 P 中只有一个后缀 m 的好后缀移动

BM 算法匹配的过程分为预处理和查找两个阶段。计算两个偏移量函数在预处理阶段完成。先定义字符集 SC:SC={x|x 在文本串中出现}，Badchar 函数计算每个字符对应的偏移量，对于每一个字符 x，可以表示为：

$$Badchar[x]=\begin{cases} t; & x\neq P[j]\ (1\leqslant j\leqslant t),\text{即 x 在 P 中未出现} \\ t-\max\{a|P[a]=x,1\leqslant a\leqslant t\}; & \text{其他情况} \end{cases} \quad (3.1)$$

当模式中某个后缀匹配成功时，可以使用 Goodsuffix 函数来计算文本指针右移的偏移量。BM 算法必须满足两个条件：

（1） tj1(i,d)：如果 d<i，则 P[i-d]≠P[i]，Goodsuffix[i+1]＝min{d>0|tj1(i,d)和 tj2(i,d)同时成立且 0≤i<t }，模式 P 的长度 用函数 Goodsuffix[0]来存储，此时 Goodsuffix 数组的空间为 t+1。

（2）tj2(i,d)：不包括后缀本身，如果要计算模式的某个后缀在最右侧开始重新出现时所对应的偏移量，对于每一个 a 必须满足 t>a，i<a<=d 或者 P[a-d]=P[a]。

Gsuff 是为了对好后缀移动 Goodsuffix 计算引入的一个数组，Gsuff[i]＝max{k|P[i-a+1…i]=P[t-a…t-1]}（当 1≤i<t 时）是对数值 Gsuff 的定义。开始查找时先将文本串的左端与模式串的左端对齐，比较则是从文本字符串与模式最右端的字符开始。文本串指针 j 向右移动，其移动的偏移量是在某次匹配不成功时计算出来的 Badchar[T[i+j]]-t+1+i 与 Goodsuffix[i]的最大值，然后将两个串按同样的方法从右向左进行逐个比较。当扫描完模式且又匹配成功，为了找出模式串在文本串中出现的位置，通过右移 Goodsuffix[0]距离的文本指针进行查找，直到文本串的末尾。

2. BM 算法流程与实例

图 3.5 是 BM 算法流程图。

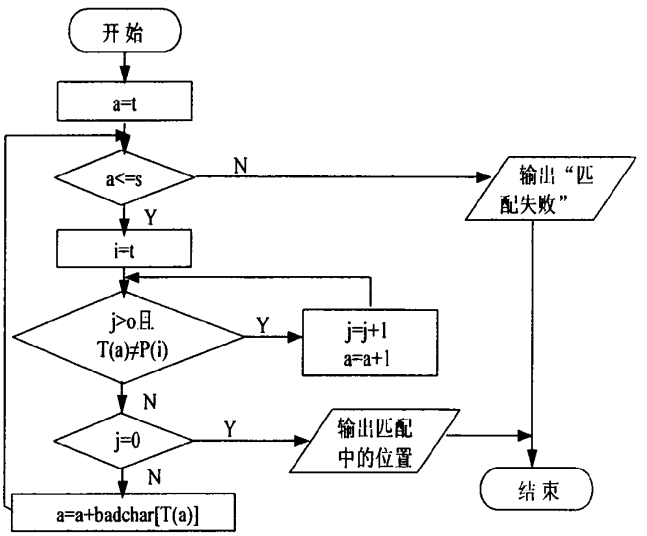

图 3.5 BM 算法流程

采用 BM 算法进行模式匹配的实例如下：模式字符串 P 为“negst”，输入字符串 T 为“jeansppknanmnegst”。按照 BM 算法，在预处理阶段得出字符串中的

字符 SC 和 Badchar 函数值的对应关系如表 3-1 所示。在匹配开始首先将文本串与模式串左端对齐，比较是从右端开始，具体匹配过程如表 3-2 所示。

表 3-1 字符集 SC 和 Badchar 函数值对照表

SC	j	e	a	n	p	k	m	g	s	t
Badchar[SC]	5	3	5	4	5	5	5	2	1	5

表 3-2 BM 算法匹配过程

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
文本串 T	j	e	a	n	s	p	p	k	n	a	n	m	n	e	g	s	t
第 1 次	n	e	g	s	t												
第 2 次		n	e	g	s	t											
第 3 次							n	e	g	s	t						
第 4 次											n	e	g	s	t		
第 5 次													n	e	g	s	t

第一次移动：i=5，SC= ‘s’，Badchar[s]=1，故向右滑动 1 个字符的位置；第二次移动：i=6，SC= ‘p’，Badchar[p]=5，故向右滑动 5 个字符的位置；第三次移动：i=11，SC= ‘n’，Badchar[n]=4，故向右滑动 4 个字符的位置；第四次移动：i=15，C= ‘g’，Badchar[g]=2，故向右滑动 2 个字符的位置；第五次时匹配成功。

3.1.2BMH 算法

BMH 算法是 Horspool 于 1980 年在论文中改进与简化了的算法，该算法是 BM 改进算法中比较出色的算法。BM 算法中采用了“坏字符”和“好后缀”两种启发规则计算在不匹配的情况下的最大可移动移位距离，但由于“好后缀”启发的规则其预处理和计算过程较复杂且难以实现。改进后的 BMH 算法在预处理阶段只使预处理函数 PBadchar，也就是模式移动时只用到“坏字符”策略。它首先比较模式串的最后一个字符和文本指针所指字符，如果相等再比较其余 m-1 个字符。只要文本串中的字符造成匹配失败，都将会由文本中和模式串最后一个位置对应的字符来启发模式向右的移动。

通过研究和实验证明：坏字符启发策略在文本串字符种类较少的情况下效率较低，为了匹配速度加快和提高匹配效率，可以通过好后缀启发策略实现。但是

在文本串字符种类较多的情况下，坏字符启发策略效率明显提高，此时好后缀启发不能明显地体现出优越性，因此通过坏字符启发大大改进了 BM 算法。BMH 算法的匹配过程如表 3-3 所示：

表 3-3 BMH 算法匹配过程

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
文本串 T	j	e	a	n	s	p	p	k	n	a	n	m	n	e	g	s	t
第 1 次	n	e	g	s	t												
第 2 次							n	e	g	s	t						
第 3 次											n	e	g	s	t		
第 4 次													n	e	g	s	t

第一次移动：模式串最后一个字符‘t’所对应的 s 的下一个字符 p 不在模式串中， 故模式串向右移动 6 个字符，从第 7 位置开始匹配；第二次移动：第 11 位的‘n’在模式串中，故模式串向右移动 4 个字符，与字符串中的‘n’对齐，从第 11 个位置开始匹配；第三次移动：第 15 位的‘g’在模式串中，为了把两个‘g’对齐，需要将模式串向右移动 2 位，此时匹配成功。

3.1.3 改进的 BM 算法

1. 算法改进思想

由于 BM 算法中两个数组的使用，增加了在预处理阶段的时间开销，但 BM 算法考虑比较全面；BMH 算法简化了 BM 算法的思想，相比而言算法更高效和简单。综合以上两种算法的优点，提出了一种基于 BM 跳跃思想的模式匹配改进算法 NBM，该算法简化了预处理的过程，加大了匹配失败后向后跳跃的幅度。经过实验验证，与原算法相比改进后的算法有效地减少了比较次数，提高模式匹配效率。

（1）匹配顺序的改进

BM 算法在匹配过程中，经常出现一些不必要的比较，这是因为出现模式的一部分后缀与文本匹配而模式的前缀不匹配的情况造成的。在改进的新 BM 算法（NBM 算法）中采用了模式两端优先比较和中间字符的策略，图 3.6 是算法的流程图。

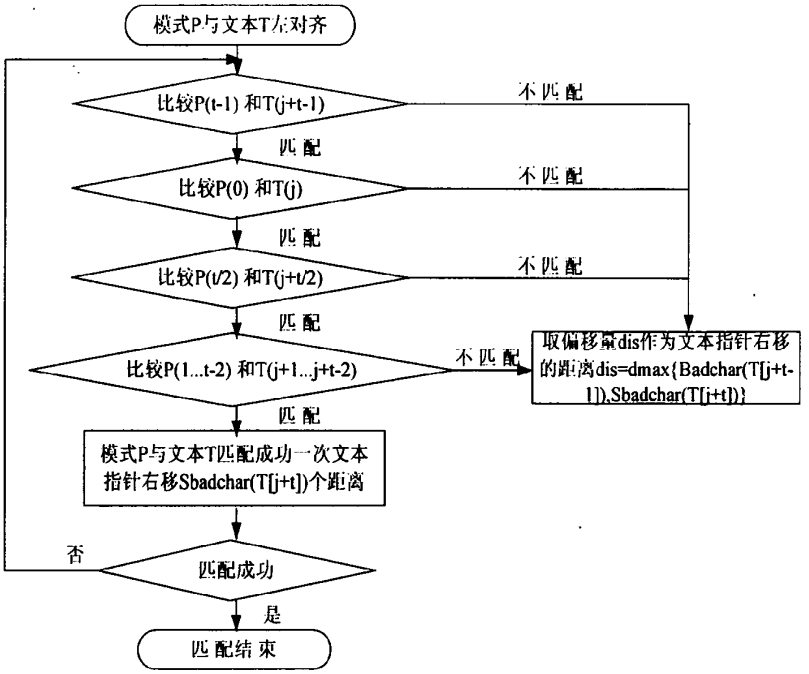

图 3.6 改进 BM 算法流程

由于很多具有相同后缀或相同前缀的规则存在于入侵检测规则库中，所以无论采用什么样的比较顺序，模式串与文本串在比较过程中都会有一些不必要的比较。NBM 模式匹配算法是一种比较适用于入侵检测系统中的算法，在该种算法中去除了一些不必要的比较，有效地减少了比较次数，提高匹配效率。

（2）坏字符启发策略的改进

假设文本窗口在文本中的位置为 T[j...j+t-1]，利用 BM 算法的坏字符移动方法在模式与文本窗口发生不匹配时，T[j+t]都将得到最大为 t+1 的移动距离，这是因为文本指针不匹配时移动距离 d 都大于等于 1。改进的坏字符移动函数 Sbadchar 表示如下，其中 x 代表字母表上的任意一个字符。

$$Sbadchar[x]=\begin{cases}\min\{i \mid 0\leqslant i<t \text{ 且 } P[t-1-i]=x\} & x \text{ 在 } P \text{ 中出现}\\ t & \text{其他情况}\end{cases} \qquad (3.2)$$

在有些情况下，使用坏字符移动的文本串，当前字符的移动距离小于其后面字符的移动距离，因此为了，我们只有取 Sbadchar[T[j+t]]和 Badchar[T[j+t-1]]两者中的较大者，才能确保每一次移动的距离最大，前面公式 3.1 给出了 Badchar[T[j+t-1]]的计算方法。

当 s 的值远远大于 t 的值的情况下，模式串在文本串中出现得就比较稀疏，此种状况下可以判断文本串中的字符出现在模式串中的可能性就相当小，t+1 的偏移量很容易被得到。

函数算法描述如下：

```
void PBadchar (int i,int Badchar[],char *K)
{
int x;
for(x=0;x<ASIZE;++x)
    Badchar[x]=i;
for(x=0;x<i-1;++x)
    Badchar[K[x]]=i-x-1;
}
void PSbadchar(int i,int Sbadchar[],char *K)
{
int x;
for(x=0;x<ASIZE;++x)
    Sbadchar[x]=i+1;
for(x=0;x<i;++x)
    Sbadchar[K[x]]=i-x;
}
void SNBM(int i,char *T,char *K,int j)
{
int y;
Int Badchar[ASIZE],Sbadchar[ASIZE];
char z,fc,mc,lc;
char *sc;
PBadchar(i,Badchar,K);
PSbadchar(i,Sbadchar,K);
fc=K[0];
```

```
sc=K+1;
mc=K[i/2];
lc=K[i-1];
y=0;
while(y<=j-i)
{
z=T[y+i-1];
if(lc==c&&fc==T[y]&&mc==T[y+i/2]&&memcmp(sc,T+y+1,i-2)==0)
    output(y);
y+=Max{Badchar[T[y+i-1]],Sbadchar[T[y+i]]};
}
}
```

2. 改进算法的性能分析

改进后的 BM 算法在最坏的情况下其时间复杂度为 O(mn)，在最优的情况下其时间复杂度为 $O\left(\frac{n}{m+1}\right)$，这一点通过以下可以证明。

通过算法的预处理阶段可以看出：每次文本指针的偏移量由于 Badchar[T[j+m-1]]和 Sbadchar[T[j+m]]的值都大于 1 的原因而必大于 1。文本中除掉后 m-1 个字符和前 m-1 个字符外，由于文本 T 与模式 P 进行匹配时每次都只能有 1 个字符跳跃，则其余字符都要进行 m+1 次的比较，显然在最坏的情况下，时间复杂度为 O(mn)。

对于不在模式中出现的字符，在预处理阶段，计算 Sbadchar，其偏移量为 m+1，在模式串 P 与文本串 T 进行匹配时每次都可以实现 m+1 个字符的跳跃，在最优的情况下，时间复杂度为 $O\left(\frac{n}{m+1}\right)$。改进后的 BM 算法在最优的情况下的时间复杂度为 $O\left(\frac{n}{m+1}\right)$，这是因为 Badchar[T[j+m-1]]和 Sbadchar[T[j+m]]二者中最大的数也在 m+1 的范围之内。

综上所述，由于 BM 算法在匹配时使用了两个数组，造成在预处理阶段时间

开销比较大，但BM算法考虑比较全面，当预处理的字母表较小时，BM算法具有较好的性能表现。改进后的BM算法当字母表相对于模式较大时其匹配效率较高，且具有较好的性能表现。

3.2 多模式匹配算法

BM算法自问世以来，就引起了各国学者的深入讨论和研究，很多BM算法的改进版本层出不穷，目的都是减少匹配过程中比较次数，尽量增加文本串指针的移动距离。但由于单模式匹配思想的BM 算法每次只能完成对一个模式的匹配工作，当然也可以实现多个模式的匹配，但对模式遍历的次数就会随模式的数量而增加，即有几个模式就需要遍历几次，效率较低。随着高速网络环境下日益增多的入侵攻击，必要要求入侵检测的规则数目也要不断增多增长，入侵检测系统中的单模式匹配算法很难满足庞大的网络数据吞吐量以及日益增加的攻击，此时人们就研究找到一种可以同时对多个模式进行匹配的算法，即多模式匹配算法。多模式匹配算法能够在一趟遍历中实现对多个模式的匹配，有效地提高了匹配速度。一般一条入侵特征可能实现很多条规则的匹配或部分匹配，通过多模式匹配可以有效地提高匹配的效率。多模式匹配算法也适用于单模式的匹配情况。

3.2.1AC 算法

贝尔实验室的Alfred V.Aho和Margaret J.Corasick于1975年提出了著名自动机多模式匹配算法，即AC算法。

AC算法其实就是一棵模式树（Tree）。对一个模式集中的一棵模式树TR的定义如下：

(1)每一条边上都用一个字符作为模式树TR的标签；

(2)不同边上的标签各不相同；

(3)若L(v)表示从根节点到v所经过的所有边上的标签的拼接，则每一个模式 p∈P都存在一个节点v使得L(v)=p；

(4)x代表任意一个叶子节点，TR中存在一个使得L(x)=p的模式p∈P。

预处理生成3个函数：goto（转移）函数，failure（失效）函数和output（输出）函数。对转移函数goto 作如下定义：

goto (s,a)：s代表当前状态， a代表边上的标签，转移函数g(s,a)表示从当前状态s开始，沿着标签a的路径所到达的状态。从状态0出发，下一个到

达的状态由当前状态和新取出的字符决定。g(0, a) =0 代表如果没有匹配的字符出现，自动机停留在初始状态。

图 3.7 是利用 goto 函数为模式集 P={do，atom，drum，doze}构造的模式树，根节点用双圈表示，其余的代表每个圆圈代表一个节点，每一条边的标签用一个字符表示，每一个模式用一个填充点圈起来。

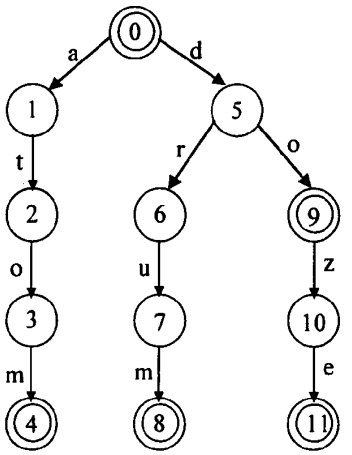

图 3.7 模式树示意图

当某个模式串与文本串匹配不成功时，应处理的下一状态有失效函数 failure 完成。对失效函数 failure 的定义如下：

Failure（s）：Failure（s）表示当 w 是某个模式的前缀且是 L(s)最长的后缀以 w 为标签的节点。

AC 算法的思想是：在 AC 算法的预处理阶段，每一个状态都对应着一个节点，这些状态有开始状态、终止状态和普通状态，开始状态用根节点表示，终止状态用叶子节点表示。把，按照规则利用转向函数和失效函数作为状态转移函数把该模式树扩展成一个树型有限自动机。

算法的匹配过程：从始状态出发，利用 goto 和 failure 函数对文本串中取出的一个字符转移进入下一状态，按照这种规则直到转移到某一个状态，若 output 函数值不为空时，此时利用 output 输出函数输出其值，表示该模式串在文本串中找到了。

AC 的有穷状态自动机 M 是由一个模式树扩展得到的，它的结构如：M=(q0，Q，Σ，g，f，F)，每一元组具体说明如下：

(1)q0∈Q，是初始状态，即根节点，该节点只用一个；

(2)Q是有穷状态集合;

(3)Σ是有穷的输入字符表;

(4)g是匹配时的转移函数;

(5)f是不匹配时的失效函数;

(6)F⊆Q，是终态集合。

自动机首先从模式树的根节点出发，沿着模式树TR的字符标签逐步向下移动，若最终沿着一条路径能够到达某一终态v，我们就说模式树TR中存在模式L(v),否则不存在。

AC算法模式匹配的时间复杂度是O(n)，与模式集中模式串的个数和每个模式串的长度都无关,并且AC算法在文本扫描时完全不产生回溯。对于TR中的每个字符标签都必须输入到有穷状态自动机中，无论在什么情况下，AC算法的时间复杂度都为O(n)。

在模式比较多的情况下,使用AC算法比使用BM算法匹配速度要提高得很多。但AC算法由于对文本只能按顺序输入，在对输入串进行搜索时无法跳过不必要的比较字符，这样就降低了AC算法实际的搜索过程中的性能。

3.2.2AC_BM算法

在多模式扫描时，使用AC算法要远比BM算法高效得很多，但AC算法在搜索是无法实现跳跃，而BM算法能够跳跃过文本串中的大段不必要的字符，使搜索速度得到了提高。许多人想到了能否利用AC算法和BM算法的优点将两者结合起来形成一种新的算法，Commentz—Walter是最先完成将BM算法和AC算法结合在一起，实验证明Commentz-Walter结合的算法要比AC算法快很多。Jang—Jong于1993年提出了利用BM算法的连续跳跃思想和AC算法的有限自动机的新算法，即AC-BM算法。在这一算法中，它首先把待查找的多个模式用一个模式树来表示，在这个模式树中，树的根节点是由具有相同前缀模式组成的。模式树的移动是从右向左的，字符比较则是从左向右匹配的。AC_BM算法同时使用好前缀移动和坏字符移动两种策略，该算法的时间复杂度为O(mn)，它利用劣势移动表和优势跳转表来实现跳跃式的并行搜索。

下面以AC_BM算法的坏字符移动为例：

设有输入文本串T=“desketocsyteyk”，模式串集P={desk，deaky，deaecr，

decdky，dectki}。模式集中最短的串是“desk”，首先将它与文本串右端对齐，如图 3.8 中的左边所示，检查则是从左向右逐个字符进行匹配。首先从“*”处开始比较，此处字符“t”与字符“d”不匹配，应采用坏字符移动策略，模式串中出现的下一个“t”在模式“dectki”中，从图中可以看出偏移量应该为 3，文本指针应该向右移动 3 个字符，如图 3.8 中的右边所示。

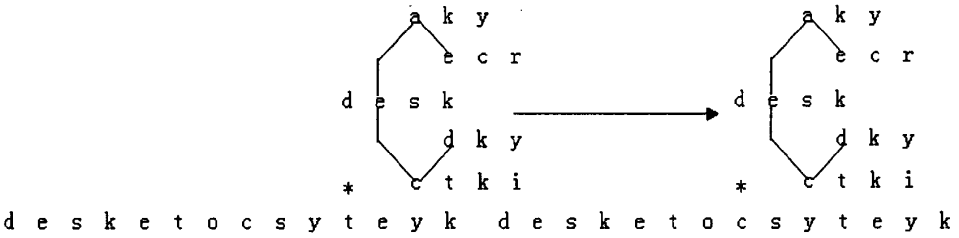

图 3.8 坏字符移动策略

AC_BM 算法的好前缀移动的例子如下所述：

设有输入文本串 T=“phdospdock”，模式串集 P={desk，deaky，deaecr，dospdo，doacpmo}，模式集中最短的串是“desk”，首先将它与文本串右端对齐，如图 3.9 中的左边所示，检测则是从左向右逐个字符进行匹配。首先从“*”处开始比较，此处文本串中字符“d”与模式中的字符“d”匹配，文本串中下一个字符“o”与其中一个模式中的字符“o”匹配，当匹配到文本串中的“c”时，在模式集找不到与它相匹配的字符，说明此时匹配失败。此时模式串“dospdo”的后缀可以看成是已匹配成功的两个字符“d”和“o”，根据好前缀移动规则，为了使文本串右边的“do” 与模式串“dospdo”右边的“do”对齐，此时需要移动 4 个字符，结果如图 3.9 中右边所示。

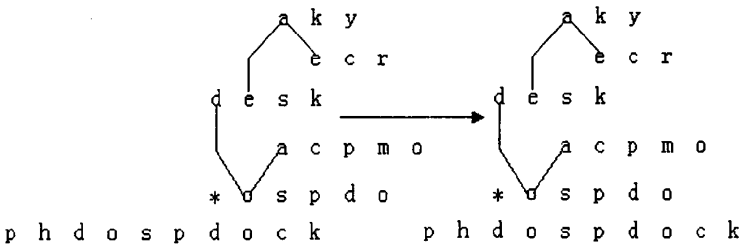

图 3.9 好前缀移动策略

当既有坏字符移动，又有好后缀移动时，则判断如果有后缀移动，就使用好前缀移动(子串移动与后缀移动的最小值)；否则使用好前缀移动与坏字符移动的最大值。

3.2.3 改进的 AC_BM 算法

1.算法改进思想

AC_BM 算法只是将 AC 算法和 BM 算法进行的结合,其跳转过程还是基于 BM 算法思想的，为了提高匹配效率，AC_BM 算法有待进一步的改进。本文提出的一种改进的 AC_BM 算法在 AC_BM 算法的基础上实现对坏字符跳跃进行改进，修改了 skip 的计算方法，在文本与模式的某次匹配失败后，跳过尽可能多的字符，实现更快的匹配过程。以下给出了 AC_BM 算法的两点改进：

第一点改进：当某次匹配失败时，T[i…i+m-1]本身使用好前缀移动对 T[i…i+m-1]左边的字符 T[i-1]使用坏字符移动，取坏字符移动和好前缀移动中的大者作为最终文本指针移动的距离，minlen 为模式集中最短模式的长度，则最大移动距离为 minlen+1。

b 代表是文本串中的任何一个字符，则坏字符移动函数定义如下：

$$skip(b)=\begin{cases}\min\{1+\min\{j|P_k[j-1]=b,0\leqslant j\leqslant minlen\},1\leqslant k\leqslant q\} & b\text{ 在 }P\text{ 中出现}\\ minlen+1 & \text{其他情况}\end{cases}\qquad(3.3)$$

第二点改进：AC_BM 算法是坏字符启发函数和好前缀启发函数的值在文本串与模式串的每一次匹配失败后都会被计算，两个函数的值以最大的作为偏移量。但在计算时好前缀启发函数所使用的时间上较长，所以我们将坏字符启发优先的策略应用到改进的 AC_BM 算法中，即能利用坏字符启发能得到最大偏移量，则好前缀启发函数就不必再计算，改进的 AC_BM 算法在模式树于文本中出现较为稀疏时，m+1 的最大偏移量叫容易在移动过程中得到，这是好前缀启发函数也不必再计算，因此使用坏字符启发优先的策略可以减少模式匹配的时间开销。

以下给出了改进后的新算法的匹配过程：

设有输入文本串 T=“desketocsyteyk”，模式串集 P={desk，deaky，deaecr，decdky，dectki}。模式集中最短的串是“desk”,首先将它与文本串右端对齐，如图 3.8 中的①所示，检查则是从左向右逐个字符进行匹配。首先从“*”处开始比较，此处文本串中的字符“t”与模式中的字符“d”不匹配，则应对“t”前面的“y”采用坏字符移动策略，根据坏字符移动规则应该将文本指针向左移动 5 个位置，也就是文本串向右移动 5 个字符，移动后的情况如图 3.9 中的②所示，此时文本串中的下一个字符“t”与模式串“desk”中的字符“d”对齐。依此类推，整个模式串与文本串的匹配过程需要 3 次移动，进行 7 次比较即可，而

采用原 AC_BM 算法要完成匹配需要移动 4 次，进行 8 次比较才可以。因此，采用改进的 AC_BM 算法的匹配效率要高于原 AC_BM 算法的匹配效率。匹配过程如图 3.10 所示。

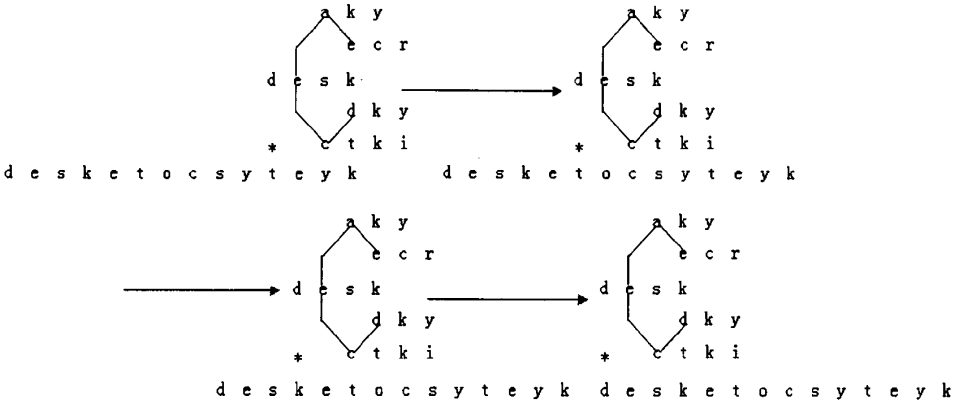

图 3.10 改进的 AC_BM 算法匹配过程

2.改进的 AC_BM 算法的性能分析

设|Σ|是字符集大小，|P|是模式集 P 中所有模式长度的总和，改进后的 AC_BM 算法其预处理时间复杂度为 O(|Σ|+|P|)。

最好的情况：每次将文本串与模式树第一个字符进行比较时都不产生匹配，产生最大的偏移量为 minlen+1，改进算法性能最好，此时的时间复杂度为 $O(\frac{n}{\min len+1})$，比较次数为 $\frac{n-\min len}{\min len+1}+\min len$。

最坏的情况：当进行文本串与模式串比较，其余字符都匹配，只有与模式最长的模式的最后一个字符比较时才不匹配，改进的算法性能最差，且此时偏移量也最小，其偏移量的值为 1，比较次数为[n-2(minlen-1)]maxlen+minlen，时间复杂度为为 O(n maxlen)。

平均时间复杂度需要通过与字符出现的概率模型计算才能得到。某一个字符不匹配要花费一定的时间，且其发生不匹配时还要进行指针后移，通过对这两者的数据进行比较我们可以考察出一个算法性能到底是好是坏。在坏字符移动上改进后的 AC_BM 算法与原算法是不同的，这里只需要比较两种算法在坏字符移动上的平均性能即可。用 Ccost(i)代表计算 skip 函数所花的代价，则改进后的算法其平均性能 $D=\sum_{i=1}^{n}\frac{C_{cost}(i)}{S_{skip}(i)}$，如果用 Pskip(i,k) 表达式表示概率，其中的 i 代表对第 i 个字符进行查找，k 代表查找不匹配后跳过的 k 个字符的，则

Sskip(i)= $\sum_{k=1}^{\min len+1} kP_{skip}(i,k)$ 为发现字符不匹配后平均跳过的字符个数，使用原 AC_BM 算法，则 Sskip(i)= $\sum_{k=1}^{\min len} kP_{skip}(i,k)$。情况下分子 Ccost(i)相同，很明显，改进的 AC_BM 算法的字符跳跃要比原 AC_BM 算法字符跳跃要大，因此改进后的算法比 AC_BM 算法具有更好的平均性能。

3.3 本章小结

本章对模式匹配算法的基本原理进行了介绍，重点分析了模式匹配算法中经典的 BM 算法、BMH 算法、AC 算法和 AC_BM 算法，对单模式匹配算法与多模式匹配算法进行了进一步研究。在单模式匹配算法中，结合 BM 算法和 BMH 算法的优点提出了一种基于 BM 跳跃思想的模式匹配改进算法。文中详细分析了 AC_BM 算法，并提出了一种 AC_BM 算法的改进算法。分别对这两种改进了的算法进行了理论分析，分析结果证明改进后的算法的确比原算法具有更好的平均性能，存在一定的优越性。

第 4 章　系统设计与分析

4.1 系统总体设计

4.1.1 评估 IDS 的性能指标

设计一个高效的入侵检测系统必须考虑多方面的因素，针对不同的网络环境采取不同的安全策略，尽量在高效与准确之间寻求一个稳定的系统。一个完善的入侵检测系统必须具有下列特点：

（1）准确性：指入侵检测系统能够检测出的入侵的准确率，在一个入侵检测系统检测不准确的情况下，极有可能产生误报和漏报的现象。

（2）可移植性：可移植性指的是该产品可以在不同的平台上运行，被多种平台支持，如 Linux、Windows 等系统。

（3）可扩展性：首先是数据与机制实现分离，也就是在机制不变的前提下能够检测出来新的攻击；其次是体系结构要具有一定的可扩展性，也就是为了保证能够检测新的攻击，在一定的条件下不对系统整体结构进行修改就能实现对检测手段的加强。

（4）实时性：在攻击发生之前发现入侵企图，入侵行为无法实时破坏，但现实的情况往往是在攻击行为正在发生的过程中入侵行为才被被检测到，此时还可以立即发出警报，作出处理。但如果是事后才发现入侵攻击，已经产生了危害，无法保证其时效性，一旦系统被攻击过，则被攻击过的系统往往就意味着后门的引入以及后续的攻击行为。

（5）安全性：一个安全系统设计的好坏，首先必须保证自身的安全，入侵检测系统也是如此，如果入侵检测系统自身的安全得不到保障，首先就意味着自身信息的无效；其次入侵检测系统一般情况下都是以特权状态运行的，很容易使入侵者控制整个入侵检测系统即获得了对系统的控制权，这种后果是不堪设想的。

4.1.2 系统模型

基于前述的入侵检测系统的功能要求，本文提出了一种基于模式匹配的入侵检测系统模型[27,28]，结构如图4.1所示。

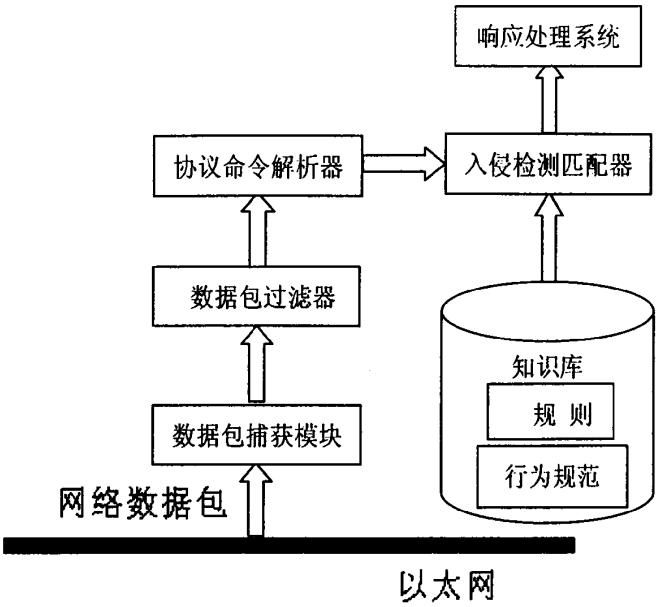

图 4.1 系统模型

该模型的工作原理是：数据包捕获模块负责从网络上获取数据包，送往数据过滤器进行数据过滤，通过协议命令解析模块中的协议字段判断各层协议，同时解析数据包的数据部分，再根据系统的知识库进行模式匹配，判断该数据包是否有入侵企图，最后交给响应处理系统对该数据包做出相应的响应处理。

4.2 具体模块设计

4.2.1数据包捕获模块

入侵检测系统检测的数据包一般来自于主机和网络，目前随着网络的普及应用，入侵行为极易存在于网络中，对网络中数据包的捕获是入侵检测的第一步。数据包的捕获[29]一般是先将整个网络上的所有信息流量先截获，之后对那些比较简单的不太关心的数据进行过滤，再将剩下的用户感兴趣的数据上传给应用程序进行分析。网络数据包的捕获有两种方法可以实现：通过监听路由器的端口和利用以太网络的广播特性实现数据的捕获，不同的情况下使用不同的捕获方式。

（1）利用路由器的监听端口捕获数据包。路由器是网络拓扑中的关键结点，它与普通的网络终端不同，如果不设置监听端口，通过规定的协议，路由器只负责对流经该路由器的数据包进行转发。通过对监听端口的设置，网络路由器除了正常的数据转发外，还可以完成数据包的复制并转发到监听端口上，达到监听的目的。数据包过滤都是在系统内核里实滤。Libpcap 是用户态下的数据包捕获 API

函数接口，支持 BPF 数据过滤机制。开发包 Libpcap 中内置了许多接口函数以及内核层实现的 BPF 过滤机制，开发包 Libpcap 的使用不仅提高了监听的效率，同时系统的开发难度也降低了很多，且增强了系统的移植性。

（2）利用以太网的广播特性捕获。以太网的数据是广播传输的，在系统正常的情况下，应用程序只能接收到本地主机上的数据包，非本地主机上的数据包则被丢弃，通过将网卡设置为混杂的模式，则流经该网段的所有数据都会被采集到，目标MAC地址的数据包，之后直接访问数据链路层，截获相关数据，不通过上层协议（诸如TCP/IP）直接由应用程序完成对数据包的过滤处理，这样就可以监听到流经本地网卡的所有数据。在UNIX系统中，可以通过以太网络数据包截获程序Tcpdump实现数据包的截获，其流程如图4.2所示。

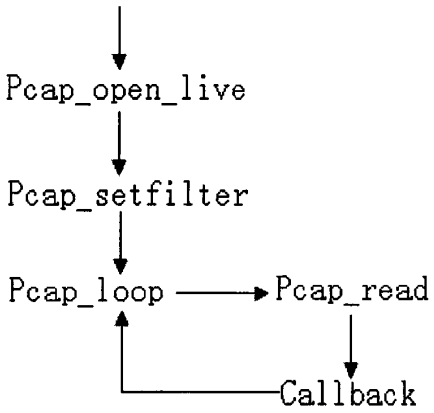

图4.2 Tcpdump流程图

本系统是利用以太网的广播特性实施数据包的捕获。

4.2.2数据包过滤模块

BPF(Berkeley Packet Filter，即伯克利数据包过滤器)是一种用于 UNIX 内核数据包的过滤器，由于一种简单的非共享的缓存模型在 BPF 中的使用，大大提高了数据包的截获性能。BPF 包括数据包过滤器和网络分接头两个部分。过滤器负责数据包的过滤，还负责被接受的数据包的那部分将被复制给应用程序；网络分解头负责完成网络设备驱动程序处数据包的收集与复制，并同时将网络设备驱动程序处数据包传递给正在截获数据包的应用程序。BPF 结构如图 4.3 所示。

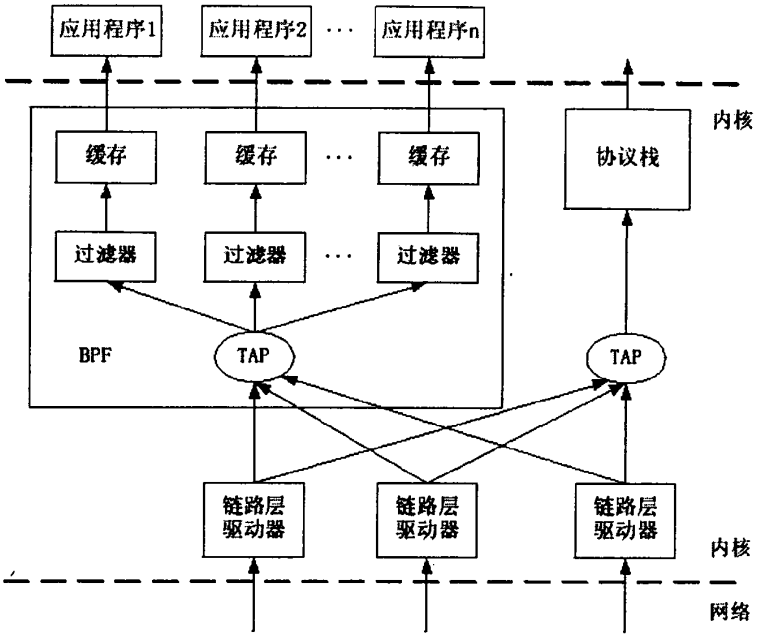

图 4.3 BPF 结构

BPF 过滤器的过滤功能由虚拟机执行过滤程序实现的，过滤机主要是由累加器、索引寄存器、数据储存器以及隐含的程序计数器组成的。过滤程序（filter programmer）实际上是一组过滤规则编写的程序，使用什么样的过滤规则完全取决于用户自己的定义，数据包是否被接受和多少个数据包被接收完全由用户的定义来决定。每一条规则执行一组操作，具体操作分为指令装载、指令储存、执行算术指令、执行跳转指令，执行返回指令等几个类别。

过滤器在过滤的过程中，一旦有数据包经过网络接口设备时，通过链路层设备驱动器将该数据包传送给协议堆栈处理。如果此时正好 BPF 也在该网络接口上截获数据，驱动器首先调用 BPF，由 BPF 负责完成传递数据包给每个监控进程的过滤器。数据包能否被接受或将哪些数据保存下来完全由这些过滤器决定。如果要接受数据包，则接受数据包的过滤器由 BPF 将需要的数据复制到与之相连的缓存中，此时设备驱动程序完成一个使命后重新获得新的控制权。被接受的网络数据包其目标地址如果不是本地地址，驱动程序在中断过程返回，如果是本地地址，驱动程序将进行正常的网络协议处理过程。

4.2.3 协议命令解析模块

协议命令解析模块[22,23]根据严格的协议规则完成通过数据包捕获模块传送过来的数据流的实时解码分析，通过网络数据报封装结构的有序性，我们可以得

到很多东西，比如该网段上运行的协议和服务、数据包的格式、数据包的源地址以及目的地址，为后期的模式匹配奠定基础。

1. TCP/IP 协议模型

TCP/IP 协议模型一般有 4 层体系结构组成，即 Internet 层、网络接口层（network interface）、传输层（transport）和应用层（application），图 4.4 中给出了 TCP/IP 协议模型的层次图。不同层上的协议完成不同的通信功能，基本上都是由下层协议为上层协调提供服务。没有下层协议，上层协议无法实现，在下层协议实现时上层协议的有些细节可以得到体现。通过对网络协议的严格分层，为下一步的协议分析提供可能。

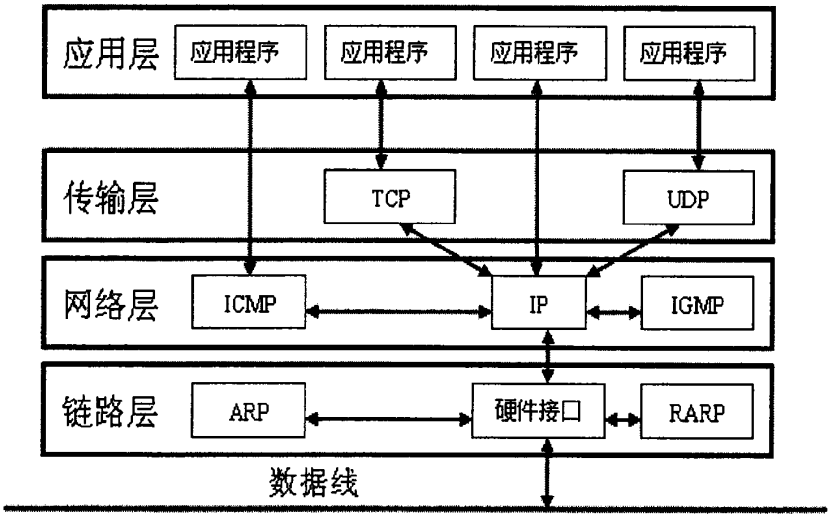

图 4.4　网络 TCP/IP 协议层次示意图

2. 协议分析结构

协议分析技术是目前和未来要研究的入侵侵检测系统探测攻击手法的技术，根据这种技术，我们可以通过网络协议的高度规则性对攻击进行快速地探测，并且可以将所有的协议构成一个四层次的链表加树的协议树结构，如图 4.5 所示。在这棵协议树中，链表的结构通过 RuleListNode 节点形成，在链表中的每一个节点都含有一个指针，通过该指针指向一棵子树，子树中的每一个节点（RuleTreeNode）代表着一个特定的协议，图 4.6 给出了关联子树结构图。且每个节点都有一个链表，链表中存储了与该节点协议相关的信息。通过链表中算法产生的进程队列中的一个进程来实现每个协议的分析，各层处理的过程基本都类似。

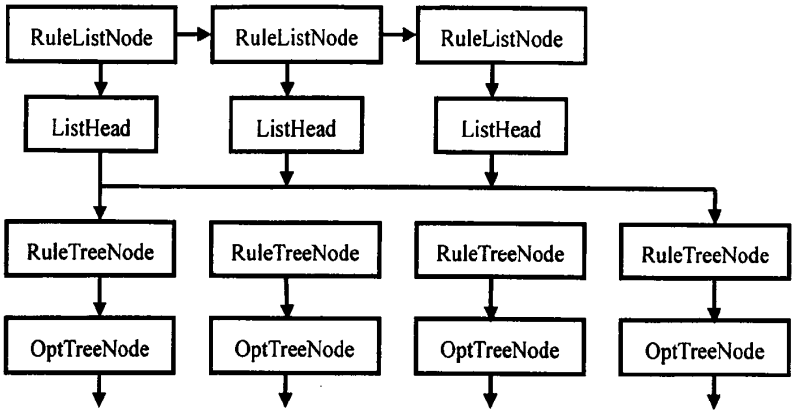

图 4.5 协议树结构图

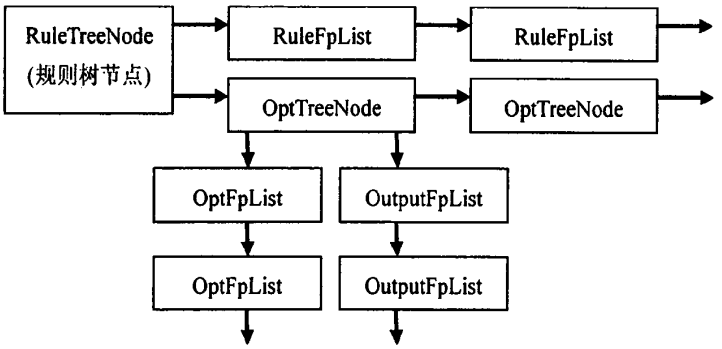

图 4.6 关联子树结构图

每条规则基本上都是有两部分组成的，包括规则头和规则体，在 RuleTreeNode 结构中就已完成了规则头的匹配。对于若干条规则其规则头相同，但规则选项部分不同的，可以把它们放在同一个 RuleTreeNode 下面的 OptTreeNode 链表中。每个 OptTreeNode 链表中都有一个 OptFpList *opt_func 链表，其中的每个节点都存放着一个选项匹配的函数。只有在链表中的所有匹配函数都成立的情况下，才能说明这条规则选项匹配成功，若使整条规则匹配，必须规则头也匹配。

协议分析技术的使用，可以使网络数据包能够及时地被分析，提高了效率，降低了系统的漏报率。命令解析的使用，特征库中只需要一种特征即可检测出所有可能的攻击串以及各种可能的攻击串的变体，提高了系统的精确性。由于协议分析能够分析出潜在的攻击行为所在的精确位置，所以使用它还可以减少简单匹配模式中存在的大量误报。在 IP 协议被分片的前提下，使用协议命令解析还可以检测分片攻击和协议验证。

4.2.4 入侵检测匹配器

从网络上捕获到的数据包经过协议解析模块解析所得到的解析数据包，通过入侵检测匹配器把该解析数据包与知识库中的规则信息进行匹配。该处使用的匹配算法采用的是改进的 AC_BM 算法，在高速网络环境下该算法能大大提高匹配的速度，由于匹配速度的提高，使得丢包现象明显减少，降低了入侵检测系统的漏报率，有效地提高了匹配效率。

4.2.5 响应处理系统

响应[7,9,30]就是当入侵检测系统检测到入侵行为时所做出的反应动作，可以分为主动响应和被动响应两种类型。

主动响应对正在进行的攻击能够阻止，使得攻击者无法继续访问，更为主动的响应则是能够对检测的攻击作出反击，但这种响应存在一定的风险，甚至会影响到网络上的无辜用户，应该谨慎采用。

被动响应则只报告和记录发生的事件，为用户提供信息，具体采取什么措施由用户决定。目前大多数入侵检测系统提供的都是多种形式生成报警的响应方式，因为这种系统能够更好地利用网络管理的基础设备，在网络管理控制台上发送和显示警报警告。

响应系统响应的措施可以分为几种：第一种措施就是收集更详细的信息；第二种措施就是修正系统堵住导致入侵发生的漏洞；第三种就是针对入侵者采取的措施，即追踪入侵者实施攻击的发起地，并采取措施以禁用入侵者的机器或网络连接。

4.3 系统网络部署

图 4.7 中给出了入侵检测系统在网络上的部署[32,33]。

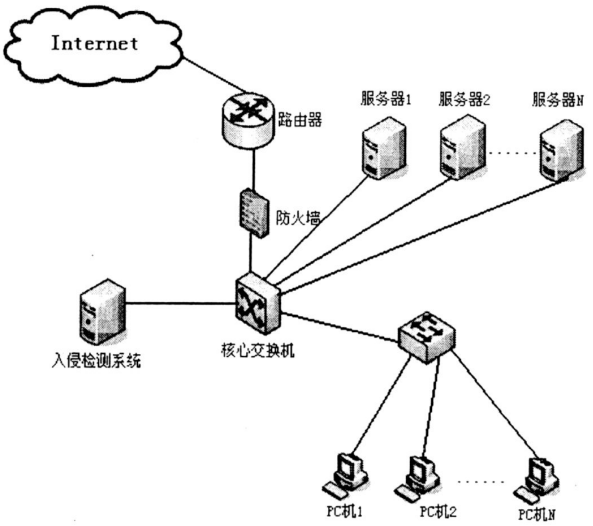

图4.7 网络部署结构图

4.4 性能测试及分析

4.4.1实验相关数据

对IDS进行测试评估，以确定检测系统能否发现数据包中的入侵。测试评估所需要的数据的生成必须靠自动完成，不需要人为的干预。实验数据必须满足具有一定的可重复性和具有一定的健壮性。测试和评估一个入侵检测系统的性能好坏，一般需要训练和实际测试两个数据，且这两部分数据中都包括有入侵数据和正常数据两种数据，为了使IDS的测试评估结果比较客观和全面，必须有正常数据的支持，入侵行为也只有在正常数据的掩护下，入侵检测系统发现入侵的机率才会大大降低。训练数据帮助入侵检测系统建立正常的行为模型，调整入侵检测系统中各参数的设置，并且训练数据中所有的入侵数据都有明确的标注。测试数据中入侵数据没有标注，它是用来进行入侵检测的。测试IDS所用的数据以实际运行环境产生的数据最为准确的，一般情况下做不到，因为实际运行环境中的入侵行为的数量根本无法满足实验检测的需要。因此在测试IDS时，很少会把IDS放在实际运行的网络中，一般需要构建一个专用的网络环境。为了测试新入侵检测系统模型的检测性能，本文以Snort做参照来比较。本实验所使用的入侵检测测试数据集是目前最常用的测试IDS的数据包之一，来源于麻省理工大学林肯实验室1999年的实验数据包作为我们的测试包。

4.4.2实验及结果分析

本文采用的入侵检测系统是基于Snort的系统模型，分别将单模式匹配算法和多模式匹配算法应用于入侵检测系统中，用户可以根据不同的需要在Snort中选择合适的模式匹配算法。本文测试所采用的算法有BM算法、改进后的BM算法、AC_BM算法和改进后的AC_BM算法，实验分为以下三组：

第一组实验测试比较基于BM算法的Snort系统和基于AC_BM算法的入侵检测系统的检测时间与规则数的变化情况，如图4.8 所示，通过该实验能够比较出两个系统存在的性能差别。

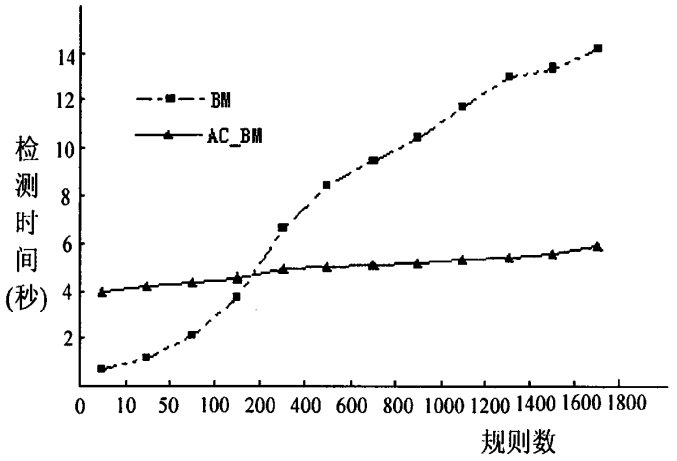

图4.8 Snort中BM算法与AC_BM算法比较

从测试结果中可以看出，由于 AC_BM 算法在生成模式树的预处理阶段占用着较多的时间，在规则数较少的情况下，BM 算法检测所花费的时间要比 AC_BM 算法少。但随着规则数的不断增加，采用 AC_BM 算法的检测时间没有明显的变化，而采用 BM 算法的检测时间则明显呈上升趋势。因此得出结论，在规则数多的情况下使用 AC_BM 算法要比使用 BM 算法其效率明显高出很多。显然规则数量大时，采用 AC_BM 算法有助于提高检测系统的性能。

第二组实验测试基于不同算法的两种入侵检测系统模型性能比较。第一种入侵检测系统采用的检测引擎是改进的 BM 算法，第二种入侵检测系统是 Snort 系统，采用的检测引擎是原 BM 算法。图 4.9 给出了这两种系统在整个数据集上检测时间与规则数的变化关系。

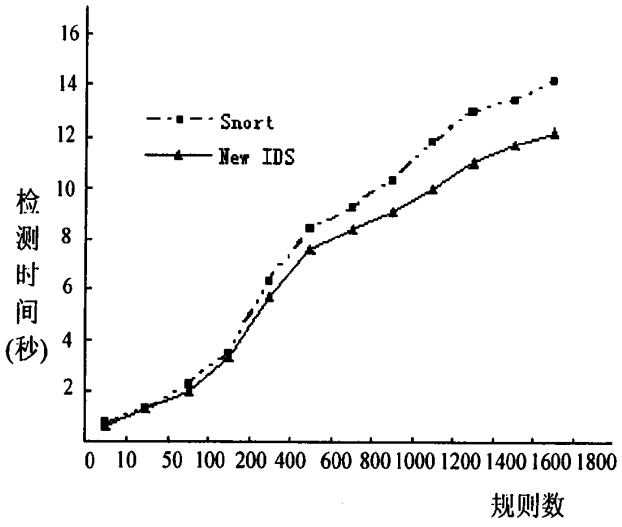

图4.9 采用原算法和改进单模式匹配算法的系统检测时间比较

从测试结果中可以看出，两种系统模型检测时间都是随着规则数的增加呈线性递增的，但是在递增的速度上基于改进的BM算法的入侵检测系统要比基于BM算法的Snort慢的多。当规则数较少时，两种系统的检测时间没有明显的差别，但随着规则数的不断增多，基于BM算法的Snort要远比基于改进的BM算法的入侵检测系统的检测时间有明显的增大，主要是因为在规则数增多时改进后的模式匹配算法采用了优先比较模式两端和中间的策略，对大量具有相同后缀的规则检测非常有效；其次，基于改进的模式匹配算法的入侵检测系统模型同时采用了规则的广度及深度搜索两种搜索策略，也对规则集进行了进一步的细化，所以，基于改进算法的入侵检测系统模型在规则数不断增多的情况下其检测速度上的优越性更加明显。

第三组实验测试的是基于改进的多模式匹配算法的检测系统和基于一般多模式匹配算法的Snort入侵检测的性能差别。一个系统检测引擎采用的是改进后的AC_BM多模式算法检测系统模型，另一个系统检测引擎采用的是AC_BM算法的SNORT入侵检测系统，图4.10给出了这两个系统在整个数据集上检测时间与规则数的变化关系情况。

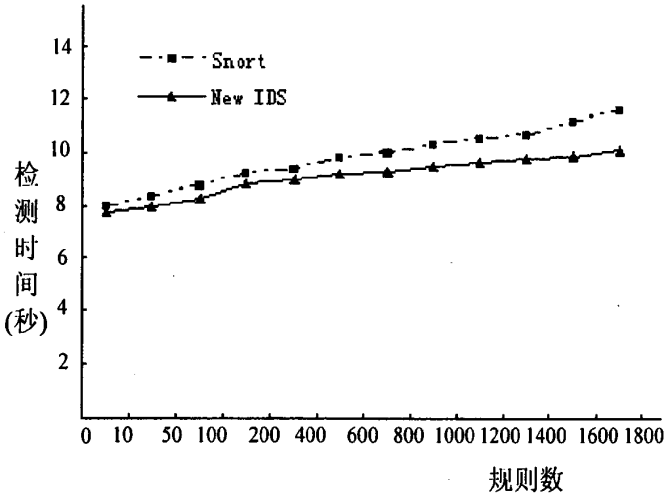

图4.10 基于多模式匹配算法的两个系统检测时间图

根据实验得出结论：基于单模式匹配算法的入侵检测系统其检测时间远远大于基于多模式匹配算法的入侵检测系统的平均检测时间。因为多模式匹配算法的预处理阶段占用的时间比查找阶段占整个检测时间要多，改进的AC_BM算法和原多模式匹配算法预处理过程相差不多，所以在规则数比较少的情况下，两种系统的检测时间差别不是很大。随着规则数的不断增多，两种系统的检测时间明显不同，因为改进的AC_BM算法在匹配过程中比原算法能够获得更大的平均偏移量，基于改进的多模式匹配算法的入侵检测系统模型的时间性能在使用了规则选项的并行搜索策略的前提下要优于基于原算法的Snort。

4.5 本章小结

本章对入侵检测系统的性能指标先做了介绍；其次针对目前入侵检测系统中存在的检测率低的问题，本章设计了一个可以提高检测效率的基于改进的AC_BM算法系统参考模型，并对主要检测部件的功能进行了详细的描述；最后用实验对采用不同算法的检测系统模型进行了测试，并进行了分析和比较。

第5章 总结与展望

随着计算机技术和网络技术的迅速发展和广泛应用，计算机安全和网络安全的重要性显得越来越突出。入侵检测作为动态安全模型核心技术之一，提供了对外部威胁、内部攻击和误操作的实时保护，是一种积极主动的安全防护技术，能够在受到危害之前进行及时的拦截和响应。

5.1 总结

入侵检测系统是一个积极主动的网络入侵防御方案。但目前的入侵检测产品检测效率低，存在着误报漏报问题，如何提高检测效率和降低入侵检测系统的误报漏报问题，是当前和未来入侵检测系统研究的热点话题。本文对传统模式匹配算法进行了分析，就单模式匹配算法中预处理阶段时间开销较大的问题，本文提出一种基于 BM 跳跃思想的改进的 NBM 模式匹配算法。NBM 算法在匹配过程中的最大偏移量均大于原算法，具有更好的平均性能，同时缩短了匹配的时间，解决了检测过程的计算量大的问题，有效地降低了误报率漏报率，提高了检测的准确率。本文还提出了一种改进的 AC_BM 算法，该算法在 AC_BM 算法的基础上实现了对坏字符跳跃进行改进，并修改了 Skip 的计算方法，在文本与模式的某次匹配失败后，跳过尽可能多的字符，实现更快的匹配过程。

本文就改进的模式匹配算法，给出了基于改进模式匹配的入侵检测系统模型，并对模型中的各模块进行了分析与设计。最后，为了测试基于改进模式匹配算法的新检测系统模型的检测性能，本文以 Snort 做参照比较，在 Snort 中根据不同的需要选择合适的模式匹配算法，用实验验证了改进后的模式匹配算法在匹配时间上明显减少，提高了检测的效率。

5.2 展望

网络入侵检测系统发展了三十多年的时间，各方面技术也取得了不小的进展，目前和未来除了完善传统的技术外，我们还需要做的是开发出高效的入侵检测技术来维护网络的安全，如主动的自主代理方法、数据融合技术、免疫学原理的应用以及智能技术等。未来的入侵检测技术大致可朝以下几个方向发展：

（1）大规模分布式入侵检测。由于传统的入侵检测技术极限性较大，一般只局限于单一的主机或网络框架，无法适应对大规模网络的监测，并且不同的入

侵检测系统之间协同性也很差。为了适应大规模网络环境下的监测，发展大规模的分布式入侵检测技术是必须的。

（2）智能型入侵检测。为了提高入侵检测系统的监测性能，将人工智能技术应用于入侵检测领域中来是相当有必要的，比如利用专家系统的思想来构建入侵检测系统。常用的有遗传算法、免疫原理、神经网络、等模糊技术方法。

（3）安全防御方案。为了充分发挥各安全产品的价值，制定出一套全面的安全防御方案是很有必要的，包括不同厂家的安全产品之间的协作，不同安全工具之间的协作，同一系统中不同入侵检测部件之间的协作，以及不同组织之间预警能力和信息的协作

参考文献

[1]丛慧源著.浅析入侵检测系统存在问题及发展趋势.电脑知识与技术，2009

[2]胡道元，闵京华著.网络安全（第2版）.清华大学出版社，2008

[3]Chuck Easttom著，张长富等译.网络防御与安全对策原理与实践.清华大学出版社，2008

[4]姚玉献.网络安全与入侵检测.计算机安全，2007

[5]戴英侠著.系统安全与入侵检测.清华大学出版社，2002

[6]胡昌振.网络入侵检测原理与技术 [M] 北京：理工大学出版社，2006

[7]唐正军著.入侵检测技术导论.机械工业出版社，2004

[8]Terry Escamilla著，施振川等译.入侵检测[M].清华大学出版社，1999

[9]杨义先，钮心忻著.入侵检测理论与技术[M]北京：高等教育出版社，2006

[10]郑成兴著.网络入侵防范的理论与实践.机械工业出版社，2006

[11]张欢，宋劲著.计算机网络入侵检测系统.中国西部科技，2008

[12]夏丹丹等.入侵检测系统综述. 网络安全技术与应用，2007

[13]李德峰著.提高分布式入侵检测系统检测检测率的研究.山东大学，2005

[14]戴玉洁著.关于网络入侵检测系统的技术改进.福建电脑，2007

[15]许占文等.一种混合式入侵检测系统的研究与设计.沈阳工业大学学报，2007

[16]刘春颂等.基于网络的入侵检测系统及其实现.计算机应用，2003

[17]黄烟波等著.入侵检测系统中误报与漏报现象研究.中国科技信息，2006

[18]赵念强等著.入侵检测系统中模式匹配算法的研究，2007

[19]彭诗力，谭汉松著. 基于特征值的多模式匹配算法及硬件实现.计算机工程与应用，2005

[20]吕秀华著.基于Snort与免疫原理混合入侵检测系统模型设计，2009

[21]韩运宝著.基于Snort的入侵检测系统的研究与改进[D].北京交通大学，2007

[22]王振东，张凤斌著.基于协议分析的分布式入侵检测系统研究.信息安全，2009

[23]凌宇著.NIDS中协议分析和模式匹配的研究.信息技术，2006

[24]刘双强，孙泽宇著.一种异常入侵检测系统误报率抑制方法.计算机应用研究，2009

[25]陈晓梅著.入侵检测中的数据预处理问题研究.计算机科学，2006

[26]张里著. 数据挖掘在网络入侵检测系统中的应用.重庆工学院学报，2008

[27]李建国等.基于数据挖掘技术的混合入侵检测模型研究.电脑知识与技术，2006

[28]李恒华等著.基于滥用检测和异常检测的入侵检测系统.计算机工程，2003

[29]金庆辉著.一种网络入侵检测中的数据包采样方法.计算机应用研究，2008

[30]邵先供等著.入侵检测响应系统的分析与研究.网络安全技术与应用，2003

[31]林果园等.通过时间同步提高分布式入侵检测系统的检测率.计算机应用与软件，2007

[32]马力波著.基于人工免疫的入侵检测与防火墙联动系统的研究与设计.中国优秀硕士学位论文全文数据库, 2008

[33]银星著.入侵检测系统应用研究.电子科技大学，2007

[34]Scott Hazelhurst,Adi Attar,and Raymond Sinnappan.Algorithms for improving the dependability of firewall and filter rule lists.IEEE Computer SocietyPress,New York,June 2000

[35]Koral Ilgun, Richard.A.Kemmerer, "State Transition Analysis: A Rule-based Intrusion Detection", IEEETransactionson S oftwareE ngineering,Vol.21N o.3p p.18 1-199(1995)

[36]Carla TL, Brodley E. Temporal sequence learning and data reduction for anomaly detection. In: Reiter M, ed. Proceedings of the 5th Conference on Computer and Communications Security. New York: ACM Press, 1998. 150-158

附录图表目录

致　谢

值此论文完成之际，首先感谢我的导师仲红老师一直以来对我的悉心指导和亲切关怀。这篇论文从选题到整个研究过程，仲红老师给了我极大的帮助，及时地为我提供了与我论文相关的信息。本课题的研究工作是在她的悉心指导下完成的。仲老师宽广的视野使我受益匪浅，还有她严谨的治学态度、深厚的专业功底、敏锐的洞察力、对工作的全身心投入和敬业精神深深感染了我，加上仲老师平易近人的工作风格，这些都将令我感动，使我终生受益。

我是一名在职攻读硕士学位的高校教学人员，工作于安徽科技学院，家庭情况比较特殊，只能是一个人一边工作一边做论文，一边还要从事繁琐的家庭事物，有时感觉真的力不从心。日常工作的繁忙使我和仲老师只能以电子邮件和电话的方式保持联系，仲老师对我的论文进展情况十分关心，多次对我的论文以电子邮件的形式进行了认真的修改，再次感谢仲红老师。

感谢安徽大学计算机学院和安徽科技学院人事处为我提供了这次进一步学习的机会，感谢一直以来给予我关心和帮助的领导、同事、同学和朋友，感谢在百忙之中审阅本文的专家教授。我的每一次进步，都离不开这些关心帮助我的人们。

请接受我最诚挚的谢意。

攻读学位期间发表的学术论文目录

1. 赵生艳，《高校校园网络安全隐患及控制策略》，农业网络信息，2009(07)146-147