摘　要

在信息网络高速发展的今天，越来越多的信息均以数字形式进行交换和管理。伴随着智能卡技术的高速发展和计算机应用的普遍推广，在校园信息管理中引入IC卡应用已成为一种趋势，IC卡的应用也正朝着由单方面应用（如食堂收费）向“一卡多用”的校园一卡通方向延伸和发展。这样，复杂的一卡通应用使其基于网络的安全问题也应运而生。

本文以南昌大学一卡通系统的网络安全为研究对象，结合网络的实际情况，通过对一卡通系统的详细介绍及网络安全的全面分析，最终选择了数据挖掘与入侵检测相结合的技术路线，采取“在线离线结合，误用异常互助，分类聚类兼顾”的思想，针对一卡通网络中关键的应用服务器提出了一种基于主机的入侵检测系统的设计方案。方案中，首先对所设计的系统框架进行了描述，然后依据框架的结构依次介绍了框架中出现的各个模块，包括数据预处理模块、数据仓库、在线分类误用检测模块和离线聚类异常检测模块等。在规则挖掘算法上，本文通过对关联算法、序列算法、关键属性、参考属性的介绍，提出了一种支持度递减滑移窗递增的层次挖掘算法，并把它作为本系统中规则挖掘的核心算法，用来对数据仓库中的标准审计数据进行规则挖掘。最后，以“口令攻击”为例，叙述了本系统进行规则挖掘与入侵检测的全过程。

关键词： 入侵检测，数据挖掘，校园一卡通，网络安全

Abstract

Nowadays, with the rapid development of information and network, more and more information is exchanged and managed in digital forms, also with the development of smart card and the generalizing of computer, using IC card in campus information management comes into a tendency. At the same time, the application of IC card has turned from single-using into multi-using, so the complicated application increased the requirement of network security.

This paper took the network security of campus card in Nanchang University as the researching object. At the beginning, it briefly introduced the campus card system and the technologies of network security, then combining the actual situation, chose the technology route that integrating data mining with intrusion detection and adopted the thinking that combining online classified misusing detection with offline clustering abnormal detection, finally aimed for the important application server of campus card system expounded a devising project of Intrusion Detection System based on mainframe. In the project, it firstly described the frame of the IDS, then specified its modules and the working principle of them, the modules included the data precondition module, data warehouse module, online classified misusing detection module and offline clustering abnormal detection module, etc. As for the algorithm of rule mining, this paper introduced the association algorithm, sequence algorithm, classification algorithm and clustering algorithm, and also included the description of key attribute and reference attribute, at last, expounded a new algorithm named layer mining of support-diminishing and sliding-window-increasing. It has been looked on as the core algorithm in this IDS and been used to dredge the standard auditing data which were stored in the data warehouse. In the end, it used "password attack" as an instance, discussed the whole running process of this IDS.

Keywords: Intrusion Detection,Data Mining,Campus Card System,Network Security

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 南昌大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：张纷婧 签字日期：2006 年 6 月 10 日

学位论文版权使用授权书

本学位论文作者完全了解 南昌大学 有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权南昌大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：张纷婧 导师签名：李建民

签字日期：2006 年 6 月 10 日 签字日期：2006 年 6 月 10 日

学位论文作者毕业后去向：上海

工作单位：中芯国际集成电路制造有限公司 电话：021-50802000-11066

通讯地址：上海市张江路 18 号 邮编：201203

第一章 绪论

1.1 问题的提出

在信息网络高速发展的今天，越来越多的信息均以数字形式进行交换和管理。伴随着智能技术的高速发展和计算机应用的普遍推广，在校园信息管理中引入IC卡应用已成为一种趋势。

自IC卡进入我国以来，各大专院校甚至中专、中学几乎都有卡在使用，广大师生在得益于卡带来方便的同时，也存在不少困扰，因为许多学校都有多种卡在使用，而这些卡系统又分别从不同的厂家独立引进，并在本部门所辖范围内各自使用。这样，由于各个系统的技术与规范不统一，造成了各种卡应用系统无法兼容，资源不能合理配置和共享的现象。学生手中少则三四张卡，多则六七张卡，给用卡带来了不便，学校也不能统一管理，比较混乱。为了解决这种纷繁复杂的局面，目前已陆续有多所高校先后迈进了校园一卡通的使用行列。

实际上，“校园一卡通”采用的是由一张校园卡代替多张IC卡的做法，在校园卡中既融合了身份识别功能又融合了金融消费功能，使得校园卡能够在“一卡通网络”中作为多张卡来使用，真正实现了“一卡在手，走遍全校；一卡通用，一卡多用”的初衷。但是，随着一卡通系统应用的逐渐深入，卡系统的用户会越来越多而且可能来自地理位置不同的区域，同时卡系统的功能也会进一步扩展以融入更多的功能，这样，必将出现一卡通系统网络规模的扩大和应用复杂度的提高，最终给网络运行的维护以及网络安全的保障提出了更高的要求。

1.2 论文的研究内容及背景

本文正是以校园一卡通系统日益提高的网络安全需求为出发点开展的一系列研究工作。

文中以南昌大学一卡通网络为研究背景，通过对网络规模及网络安全的调查分析发现，我校在校园一卡通项目上选择的是增加投入换取安全减小风险的做

法，采用专网搭建。在外网方面，只有通过路由器与建行的连接，没有与 Internet 的互联，所以在与外网隔离方面有一定的安全保障。但是，据有关调查报告显示，世界上所有的网络攻击中有高达 70%的攻击都来自于网络内部(如内部心怀不轨的员工所进行的非法操作等)，正所谓信任是一切安全的基础，因此网络内部的安全问题是绝对不能忽视的。这样，考虑到一卡通网络内部数据的重要性，在网络内部采取安全措施是极为有必要的。这也正是本文研究的关键问题。

在对我校一卡通系统的网络规模及网络安全进行充分分析后，本文提出了一套有针对性的基于数据挖掘的主机入侵检测系统设计方案，选择数据挖掘与入侵检测相结合的技术路线，采取"在线离线结合，误用异常互助，分类聚类兼顾"的思想，目的是保障内部网络更加安全可靠的运行。

实际上，将数据挖掘技术应用到入侵检测中，是目前入侵检测领域一个比较重要的研究方向。众多实验和测试结果表明，该做法在理论上是可行的，在技术上也是可能的。数据挖掘是一种特定应用的数据分析过程，可以从包含大量冗余信息的数据中提取出尽可能多的隐藏知识，从而为作出正确的判断提供基础。将数据挖掘技术应用到入侵检测中，可以用适当的挖掘算法来对海量的安全审计数据进行智能化处理，自动系统地抽象出利于进行判断和比较的特征模型（可以是基于误用检测的特征向量模型，也可以是基于异常检测的行为描述模型)，进而自觉地维护入侵检测系统的特征模式库，这样的入侵检测系统将具备良好的自适应性和可扩展性。本文所做的正是这方面的尝试性研究。

1.3 论文的研究意义

论文的研究意义可以从以下几方面来叙述：

首先，对于目前我校一卡通系统而言，由于本文拟解决的问题正是一卡通系统的网络安全，所设计的基于主机的入侵检测系统也正是以此为出发点的，所以论文的研究对提高一卡通系统的网络安全有一定的实际意义。

其次，对于今后我校一卡通系统的发展甚至其他企业院校中的一卡通系统而言，在网络安全方面提供了一定的依据及参考。

再次，对于入侵检测系统的进一步发展而言，由于本文所研究的是结合数据挖掘技术的基于主机的入侵检测系统，它一方面将数据挖掘技术运用到了基于主

机的入侵检测系统中，另一方面以 Windows 自带的事件查看器作为审计数据的来源，这在基于主机的入侵检测技术领域中具有一定的参考价值。

最后，对于数据挖掘技术而言，本文在对多种数据挖掘算法充分理解的基础上，针对校园一卡通系统中应用服务器的审计日志提出了一种支持度递减滑移窗递增的层次挖掘算法，该算法能够较全面的对数据进行规则挖掘，对于应用在入侵检测系统中的数据挖掘算法而言有一定的参考意义。

1.4 论文内容编排

本文的结构以如下的方式进行组织，全文共由五章组成：

第一章为绪论，引出了本文研究的问题，并简单介绍了论文的研究内容、背景及意义，最后列出了论文的具体编排结构。

第二章为校园一卡通简介，首先对一卡通做了整体介绍，然后针对南昌大学校园一卡通的具体情况，先后从网络规模、功能、软硬件系统几方面分析了一卡通的现状，并在此基础上全面分析了我校一卡通的网络安全。

第三章为基于数据挖掘的入侵检测技术，详细介绍了本文所采用的相关技术，从对网络安全的简要介绍开始，依次介绍了入侵检测技术和数据挖掘技术，并说明了基于数据挖掘的入侵检测技术的研究现状。

第四章为系统详细设计，是本文的核心章节。它首先对所设计的基于主机的入侵检测系统框架进行了描述，然后依据框架的结构依次介绍了框架中出现的各个模块。在规则挖掘算法上，提出了一种支持度递减滑移窗递增的层次挖掘算法，并把它作为系统的核心挖掘算法，用来对系统中的审计数据进行规则挖掘。

第五章为实验设计与讨论，以“口令攻击”为例，叙述了该系统规则挖掘与入侵检测的全过程。

第六章为结束语与展望，总结本论文所做的工作，并对进一步的研究工作进行展望。

第二章 校园一卡通

作为本文的研究对象，本章对校园一卡通做了系统的介绍。首先宏观上给出了校园一卡通的定义，然后针对南昌大学校园一卡通的具体情况，结合网络拓扑示意图，先后从网络规模、功能、软硬件系统几方面分析了一卡通的现状，并在此基础上从链路传输、网络结构、操作系统和应用系统四方面详细地分析了我校一卡通的网络安全。

2.1 校园一卡通简介

2.1.1 校园一卡通的概念

根据多年对各种卡的探索、研究及智能卡管理系统工程的开发与运用，有关专家总结出真正的“一卡通”概念：即“一卡一库一线”。

所谓“一卡”，就是在同一张卡上实现对多种不同功能的统一智能管理，一张卡同时适应多种设备（含读写设备），具备多种功能，满足多种应用。

所谓“一库”，就是在同一个软件系统（一卡通软件）下操作同一个数据库（一卡通数据库），实现卡的发行、取消、报失、消费、查询等多种功能。

所谓“一线”，就是多种不同的设备挂在“一条线”（一卡通网络）上，进行不同数据的信息交换，也就是多种不同的设备都能够通过网络互联。

“校园一卡通”属于“一卡通”系统应用的具体项目之一，是以 IC 卡为信息载体，通过校园网络来实现金融消费功能和信息管理功能的综合系统。它以 IC 卡取代了学校管理中的各种个人证件和校园生活中的各种支付手段，将学生的就餐管理、图书管理、就医管理、考试管理、学分管理、机房管理、收费管理、身份管理及其他消费管理以 IC 卡为媒介，来统一实现。

2.1.2 校园一卡通的作用和应用范围

1、校园卡在校园内可代替：

(1) 身份卡：学生证（工作证）、借书证、会员证、准考证、出入证等；

(2) 现金卡：电子钱包、电子存折、就餐卡、上机证、医疗证等。

2、“校园一卡通”的应用范围：

(1) 管理功能（身份认证）：学生管理、教务管理、图书馆管理、机房管理、考勤管理与考试管理等。

(2) 消费功能（电子钱包）：（水电费、上网费、电话费、学费、上机费等）交费、（食堂）就餐；（零售商店）购物、（浴室）洗浴；（医务所）就医；（体育馆、自行车库、水房等）消费。

(3) 财务功能（结算中心）：学校财务中心财务集中处理及与银行的对帐处理；与银行之间的转帐功能，包括“校园一通卡”与银行个人帐户间的圈存处理等。

(4) 自助功能（自助服务）：教职工与学生持卡自助查询电子钱包（校园帐户）余额、银行帐户余额、学生成绩、学分等信息；学生可持卡选课、电子钱包圈存、存款、取款、汇款等。

2.1.3 校园卡系统现状分析

自 IC 卡进入我国，各大专院校甚至中专、中学几乎都有卡在使用，广大师生在得益于卡带来方便的同时，也存在不少困扰，许多学校都有多种卡在使用，这些卡系统分别从不同的厂家独立引进，并在本部门所辖范围内使用；由于各个系统的技术与规范不统一，造成了各种卡应用系统无法兼容，资源不能合理配置和共享。学生手中少则三四张卡，多则六七张卡，给用卡带来了不便。学校也不能统一管理，比较混乱。为了解决这种纷繁复杂的局面，目前已陆续有许多高等院校迈入了校园一卡通的使用行列。

2.1.4 校园一卡通的发展趋势

校园一卡通的发展趋势可以从以下几方面进行考虑：

首先，日益普及的高校校园网为一卡通系统提供了网络基础；

其次，卡片应用技术（硬件和软件两方面）的逐渐成熟为一卡通系统提供了技术基础；

第三，各学校混乱的卡系统对一卡通系统提出了现实需求；

最后，银行方面的积极参与为银校一卡通系统的建设提供了动力支持。

因此校园一卡通是今后校园卡发展的必然趋势。

2.1.5 校园一卡通系统的价值

校园一卡通发展的必然趋势也可以从另一个侧面来体现，那就是一卡通系统多方面的使用价值：

(1) 减少校内执行收费和身份认证的工作人员，提高工作效率，方便广大师生的日常生活；

(2) 减轻了系统维护人员的劳动强度，变对多种不同系统的维护为对单一统一系统的维护；

(3) 部分自动化系统实现无人监管、自助消费的方式，可以根据需要适当延长开放时间，甚至实现 24 小时不间断开放；

(4) 提高对校园网设备的利用率，并在一定程度上减小了学校信息化建设的使用经费。

2.2 我校一卡通网络的现状

2.2.1 网络规模及功能

为提高南昌大学信息化水平，方便广大师生员工的工作、学习和生活，我校已采用“校园一卡通”网络系统。目前校园一卡通只推行于前湖新校区，实现的功能主要有餐饮、图书借阅、上机、购水、购电、就医等，最近还新增了考勤与无线车载功能。

考虑到校园网络本身的复杂性以及与互联网相连的互通性，如果将一卡通系统搭建在校园网络之上将存在过多的安全隐患，需要全面系统有针对性的安全措施来加以保护（可能涉及到 VPN、防火墙、入侵检测等多种网络安全技术的复用），这一方面给安全系统的部署提出了高要求，另一方面也增加了校园网络的负荷，影响到校园网络本身的可靠运行。因此我校选择了增加投入换取安全减小风险的做法，采用专网搭建，避开了与校园网络的设备共用，一定程度上减小了风险系数，保障了系统的可靠运行。

随着学校的不断发展，校园一卡通的规模也将进一步扩大，包括逻辑上功能的扩充和物理上区域的拓展，以求逐步实现在各校区全方位多功能的互连互通，真正做到“一卡在手，走遍全校；一卡通用，一卡多用”。

2.2.2 网络架构

2.2.2.1 网络拓扑示意图

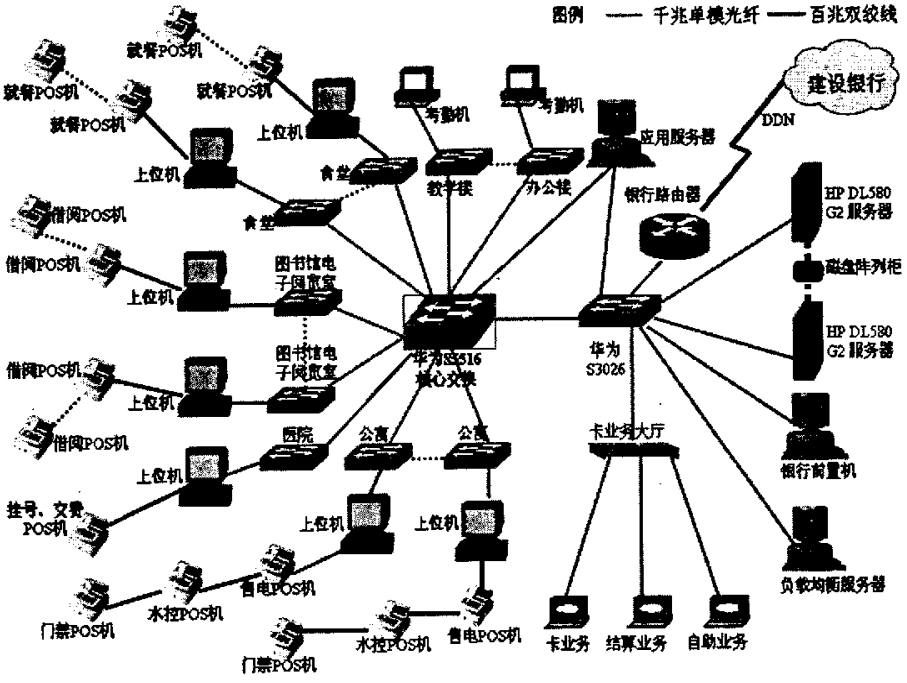

图 2.1 南昌大学一卡通网络拓扑示意图

如图 2.1 所示，目前一卡通网络的拓扑情况由下至上依次为：终端设备由 485 线路上连至上位机；上位机由 485 线路连接至接入层交换机，应用服务器和数据库服务器也由 485 线或光纤连接至接入层交换机；再由接入层交换机经光纤连接至汇聚层交换机再连接至中心交换机或直接连至中心交换机；与建行网络互联的路由器则连接在和应用服务器及数据库服务器相连的同一个接入层交换机上。

2.2.2.2 网络的硬件系统

一卡通网络由以下几类硬件搭建而成：

1、终端 POS 机：

是读卡的终端设备，采用 485 通讯线路连接，所有刷卡消费和身份识别的动作都由终端设备来完成。

2、上位机：

是控制终端设备的计算机系统，由计算机、控制器、485 线组成，所有刷卡的消费记录或认证记录及相关数据的采集和上传都由这类机器完成。

3、应用服务器：

是一卡通软件系统的服务器，由 DELL 服务器构成，包括一卡通的 Web 服务器和银行前置机，其中 Web 服务器是客户端与数据库的中间设备，负责一卡通系统的管理维护及上位机数据的接收处理，而银行前置机则负责银行圈存及交易处理等。

4、数据库服务器：

数据库服务器是整个一卡通系统的数据核心，由两台 HP 服务器构成，负责存储所有一卡通系统的相关数据。

5、交换及路由设备

交换机（包括接入层交换机、汇聚层交换机和中心交换机）负责连接内网中分散在不同地域的各类设备，路由器负责连接外网的建行服务器。

2.2.2.3 网络的软件系统

1、上位机 POS 控制软件：

运行在 Windows 2000 Server 操作系统上的控制软件，负责把终端设备的数据实时地显示和传输到上位机上，它必须保证正常运行，由它控制的终端设备才能工作。

2、后台应用服务：

(1) 一卡通管理应用系统

运行在 Windows 2000 Server 操作系统上的应用软件，负责卡务中心业务办理及系统设置，是一卡通系统应用功能的操作平台。

(2) 中间件应用系统

运行在 Windows 2000 Server 操作系统上的应用软件，负责与数据库及相关后台的应用实现，管理上位机数据的传输、整理、应用功能的逻辑实现、银行圈存等。

3、数据库系统：

运行在 SCO UNIXWARE 7.1.1 操作系统上的 ORACLE 8.1.7 数据库软件，负责校园卡所有用户信息及消费数据的备份、查询、添加、删除和修改等。

2.3 网络安全分析

如前所述，对于校园一卡通系统我校选择的是增加投入换取安全减小风险的做法，采用专网搭建。由于是专网，所以安全性相对较好，但是说到网络安全从来只有相对的概念而没有绝对的说法。下面分别从链路传输、网络结构、操作系统和应用系统四方面来对现有一卡通网络进行全面的网络安全分析。

2.3.1 链路传输的安全分析

一卡通的数据传输是一卡通终端通过一卡通专网先后经由485双绞线、千兆单模光纤历经上位机和交换机传送到数据库服务器的。

对于链路中传输数据的入侵方式存在两种情况：一种情况是入侵者直接到内部网上进行攻击、窃取或其它破坏；另一种情况是入侵者在传输线路上安装窃听装置，窃取网上传输的重要数据，再通过一些技术读出数据信息，造成泄密或者做一些篡改来破坏数据的完整性。针对链路风险通常采用的安全防护措施是对传输的数据加密，并通过数字签名及认证技术来保证数据在网上传输的真实性、机密性、可靠性及完整性。

然而，由于在一卡通网络中传输的数据大都是用户信息及消费数据，机密性相对较低，因此这并不是保证一卡通网络安全运行所需解决的首要问题。

2.3.2 网络结构的安全分析

1、与外网连接的安全威胁

在外网方面，一卡通网络只有通过路由器与建行的连接，没有与Internet的互连，在考勤与图书借阅功能中存在与校园网络数据的“关联”（物理上是隔开的），会定期将有关数据导入到一卡通系统数据库中，使得通过校园网能够访问和查询这些数据，但目前一卡通网络还不能访问校园网。这样，真正与外网有连接的就是与建行网络的连接了，而建行网络本身的安全性是毋庸置疑的，所以来自外部网段的攻击出现的可能性相对较小。

通常，针对外网攻击采用的安全防护措施是加设防火墙，但根据目前网络的状况，出现外部攻击的可能性较小，所以目前可以暂不考虑部署防火墙设备来阻

止来自外部的攻击。

2、内部网络的安全威胁

据调查在已有的网络安全攻击事件中约70%是来自内部网络的侵犯。比如内部人员故意泄露内部网络的网络结构；安全管理员有意透露其用户名及口令；内部不怀好意人员编些破坏程序在内部网上传播或者内部人员通过各种方式盗取他人涉密信息传播出去等。种种情况都可能对网络安全造成严重威胁。

对于目前一卡通网络，由于用户众多，身份复杂，包括各个校区的老师、学生、员工、家属等等，所以网络内部的安全问题是目前一卡通网络安全的首要问题。正所谓信任是一切安全的基础，即使与外部网段隔离的再好，网络内部的安全问题也是绝对不能忽视的。

通常，针对内部网络运用最普遍的安全防护措施就是采取入侵检测技术。入侵检测分为基于主机的入侵检测和基于网络的入侵检测，依据一卡通网络的具体情况，由于网络较简单且基本没有与外网的连接，所以应该重点考虑基于主机的入侵检测。

2.3.3 操作系统的安全分析

操作系统安全通常是指网络操作系统的安全。目前的操作系统无论是Windows还是其它任何商用UNIX操作系统，系统本身必定存在安全漏洞。这些安全漏洞都将存在重大安全隐患。

对于一卡通网络，所用到的操作系统有 Windows 2000 Server 和 SCO UNIXWARE 7.1.1，不可避免它们都会存在安全漏洞，但这种安全问题只能通过操作系统的生产厂家进行不断地完善，对于用户是很难对其加以改进的。

2.3.4 应用系统的安全分析

应用系统的安全涉及很多方面。应用系统是动态的、不断变化的，应用的安全性也是动态的，这就需要对不同的应用，检测安全漏洞，采取相应的安全措施，降低应用的安全风险。

对于一卡通网络，所涉及的应用系统有上位机控制软件、一卡通管理应用系统、中间件应用系统和ORACLE 8.1.7数据库系统等。同样它们的安全也应该由

开发厂商来保证，用户所能做的只是按照要求进行合理的运用与维护。

2.3.5 网络安全小结

上述分析表明，目前的一卡通网络，在网络安全方面所需解决的首要问题应该是网络内部的安全问题，本文将紧扣这个问题，采取基于主机的入侵检测技术与数据挖掘技术相结合的路线进行研究与探讨。

2.4 本章小结

本章中首先对一卡通做了整体介绍，分别叙述了校园一卡通的概念、作用、应用范围和现实意义。然后针对南昌大学校园一卡通的具体情况，结合网络拓扑示意图，先后从网络规模、功能、软硬件系统几方面分析了一卡通的现状，并在此基础上从链路传输、网络结构、操作系统和应用系统四方面详细地分析了我校一卡通的网络安全，最后得出结论，即所需解决的网络安全的首要问题是网络内部的安全问题。

第三章　基于数据挖掘的入侵检测技术

由于本文的研究对象是南昌大学一卡通网络的安全，通过第二章对网络安全的分析，我们知道所需解决的首要问题是网络内部的安全问题，而解决网络内部安全问题最适用的方法就是入侵检测技术，根据网络的具体情况及规模，我们进一步明确应该采用基于主机的入侵检测技术。同时，为了提高入侵检测系统的有效性、适应性和扩展性，本文考虑将数据挖掘技术结合到入侵检测技术中，来对规则库进行动态维护。

因此，作为本文研究的基础，本章将对相关领域知识进行介绍，从对网络安全的简要介绍开始，依次介绍入侵检测技术（着重介绍基于主机的入侵检测技术）和数据挖掘技术，最后说明了数据挖掘技术在入侵检测领域的应用状况。为论文的进一步深入提供了理论基础。

3.1 网络安全介绍[28][24]

3.1.1　网络信息安全的定义

网络信息安全的定义[20]有很多种，国际标准委员会 ISO 定义为："为数据处理系统采取的技术和管理的安全保护，保护计算机硬件、软件、数据不因偶然的或恶意的原因而遭到破坏、更改和泄露"。我国公安部计算机管理监察司的定义是："计算机安全是指计算机资产的安全，即计算机信息系统资源和信息资源不受自然和人为有害因素的威胁和危害"。

针对网络信息安全的相对性，著名的网络安全公司安氏也给出了一个通俗的、动态的定义："网络的安全实际上是理想中的安全策略和实际的执行之间的一个平衡"。现实中并没有一种技术可以完全消灭网络安全中的漏洞，正如 P2DR（Policy 策略、Protection 防护、Detection 检测、Response 响应）安全理论模型[2]中指出的安全目标：尽可能的增大保护时间、尽量减少检测时间和响应时间。

3.1.2 网络安全的目标

从防卫的角度来看，网络安全的目标[17][40]包括以下几个方面：

(1) 网络服务的可用性(Availability)：无论何时，网络服务必须是可用的，如抗击拒绝服务攻击。

(2) 网络信息的保密性(Confidentiality)：网络服务要求能防止敏感服务信息泄露，要求只有在授权的条件下，才能获取服务信息。

(3) 网络信息的完整性(Integrity)：网络服务必须保证服务者提供的信息内容不能被非授权篡改，不管是有意或无意，完整性是对信息的准确性和可靠性的评价指标。

(4) 网络信息的非否认(抗抵赖)性(No-repudiation)：用户不能否认消息或文件来源地，也不能否认接收了信息或文件。

(5) 网络运行的可控性(Controllability)：是指网络管理的可控性，包括网络运行的物理的可控性和逻辑或配置的可控性等，能够有效地控制网络用户的行为及信息的传播范围。

3.1.3 网络安全技术的分类

网络安全技术[3]总体看来，可划分为两种：静态安全技术和动态安全技术。

静态安全技术包括防火墙技术、VLAN 技术、加密技术和身份验证技术等。

其中防火墙是静态安全技术中最常用也是最成熟的技术，设置在内外部网络之间，通过监测、限制、更改跨越防火墙的数据流，尽可能地对外部屏蔽内部的信息、结构和运行状况，阻止不可预测的、潜在破坏性的侵入，来保护内部网络不受外部影响。但是防火墙只能实现网络的隔离，不能检查经过它的合法流量中是否包含恶意入侵。

动态安全技术包括入侵检测技术和陷阱网络技术等。

其中入侵检测是动态安全技术中最成熟也是最有代表性的技术，它通过不断检测和监控网络系统来发现新的威胁和弱点，然后反复循环反馈来及时做出有效的响应，因此能够主动检测网络的易受攻击点和安全漏洞，并且通常能够先于人工探测到危险行为。但是入侵检测只能分析数据记录对入侵做出判断并进行反

馈，不能直接对其进行响应。

通过上述介绍，我们可以发现静态安全技术和动态安全技术各有其长处和不足，因此在实施网络安全策略的时候，应该结合网络的具体情况采取适当合理的安全措施，有必要时可以考虑将静态和动态两者结合起来互动运行。

然而通过第二章对我校一卡通系统所作的网络安全分析，我们知道目前南昌大学的一卡通网络是专网且基本没有与外网的连接，所以可以暂不考虑静态安全技术中的 VLAN 技术与防火墙等技术，而应重点考虑动态安全技术中的入侵检测技术，尤其是基于主机的入侵检测技术。下面重点介绍入侵检测技术。

3.2 入侵检测技术[28][38]

3.2.1 入侵检测[9]概述

在介绍入侵检测之前，我们先引出入侵的概念。入侵指的是破坏目标系统资源的完整性、机密性或可用性的一系列活动。

这样入侵检测可以简单定义为对(网络)系统的运行状态进行监视，发现各种攻击企图、攻击行为或者攻击结果，以保证系统资源的机密性、完整性与可用性的行为。入侵检测对计算机和网络资源上的恶意使用行为进行识别和响应，它不仅检测来自外部的入侵行为，同时也检测内部用户的未授权活动。

入侵检测系统(IDS)是从计算机网络系统中的若干关键点收集信息，并分析这些信息，检查网络中是否有违反安全策略的行为和遭到袭击的迹象。IDS 所检测的入侵不包括物理入侵，而仅包括以电子方式从系统内部或者系统外部发起的，尝试或者实施对系统资源的非授权访问、操纵或破坏的行为。IDS 自动收集并分析审计数据，一旦发现入侵迹象，则采取适当措施(如报警，断开相应的连接等)，保护系统不被攻击。

3.2.2 入侵检测系统的分类及原理[35]

3.2.2.1 入侵检测系统的分类

对入侵检测系统的分类[8]方法很多，表 3.1 从不同的角度对其进行了分类。

表 3.1 入侵检测的分类

分类的角度	数据来源			分析方法		时效性		分布性	
类型	基于主机	基于网络	混合型	异常检测	误用检测	离线分析	在线分析	集中式	分布式

3.2.2.2 入侵检测系统的原理[38]

入侵检测系统的原理主要就是检测技术的原理，可以从以下两方面分析：

(1) 异常入侵检测原理

异常检测指的是根据系统或用户的非正常行为和使用计算机资源的非正常情况检测出入侵行为。异常入侵检测假定所有入侵行为都与正常行为不同。如果建立目标系统(受监控系统)及其用户，那么理论上可以把所有与正常活动轮廓不同的系统状态视为可疑活动。对异常阀值与特征的选择是异常入侵检测的关键，预先定义固定的限制范围，如果考察的值超出了这个范围，那么就怀疑有入侵。

(2) 误用入侵检测原理

误用入侵检测是指根据已知的入侵模式来检测入侵行为。误用检测假设所有入侵方法(及其变种)都能被精确表达为一种模式或特征，而所有已知的入侵行为都可以通过模式匹配来检测。误用入侵检测的关键是如何表达入侵的模式，把真正的入侵与正常行为区分开来。

(3) 误用检测与异常检测的比较

基于异常和基于误用两种不同的检测方法，横向比较一下其区别在于：

异常检测系统试图发现一些未知的入侵行为；而误用检测系统则是标识一些已知的入侵行为。

异常检测指根据使用者的行为或资源使用状况来判断是否入侵，而不依赖于具体的行为是否出现来检测；而误用检测系统则大多是通过对一些具体的行为的判断和推理，从而检测出入侵。

异常检测的主要缺陷在于误检率很高，尤其在用户数目众多或工作行为经常改变的环境中；而误用检测系统由于依据具体特征库进行判断，准确度要高很多。

异常检测对具体系统的依赖性相对较小；而误用检测系统对具体的系统依赖性太强，移植性不好。

3.2.3 基于主机的入侵检测系统[33]

当系统用于分析计算机（主机）产生的数据（例如应用程序及操作系统的事件日志）时，入侵检测就是基于主机的。与之相对的是基于网络的入侵检测，它用于处理来自网络上的数据（例如 TCP/IP 通信量）。尽管网络入侵检测应用得很广泛，但是主机入侵检测由于内部人员的威胁正变得越来越重要。

因为目标数据源临近已通过验证的用户，所以基于主机的入侵检测对于检测内部人员的误用特别有效。主机事件日志包含的信息说明了已验证（或内部）用户访问特定文件及执行程序的情况。这种入侵检测也能提供很好的毁坏情况评估数据。如果保护得好的话，必要时事件日志可以在法庭上支持对计算机罪犯的起诉。

3.2.3.1 基于主机的入侵检测结构

基于主机的入侵检测系统有两种结构：集中式结构和分布式结构。

(1)集中式基于主机的入侵检测结构

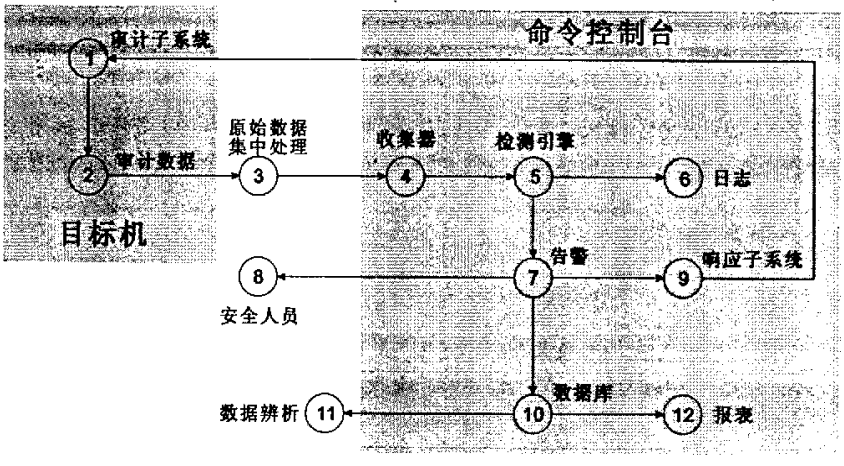

图 3.1 集中式基于主机的入侵检测结构

如图 3.1 所示，是集中式基于主机的入侵检测结构，所谓集中就是将多个目标机的审计数据集中在同一个命令控制台的分析引擎中统一检测。图中演示了一个事件记录在该结构中的生命周期。

根据集中式结构的特点，我们可以总结出它的优缺点：

优点在于：一，对目标机的性能影响很小或没有影响，因为所有的分析都是在控制台进行的，这就允许进行更复杂的检测。二，可以实现多主机联合检测，

因为集中式引擎可以访问来自所有目标机的数据。三，如果集中的原始数据保护的好的话，必要时可以将这些数据用在起诉中。

缺点在于：一，如果目标机的数量大或中央检测引擎速度慢的话，不能进行实时检测或实时响应。二，将大量原始数据集中起来会影响网络通信量。

(2)分布式基于主机的入侵检测结构

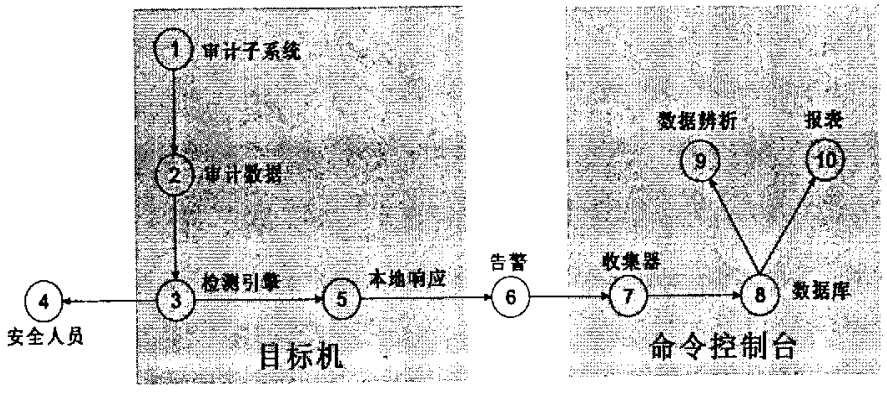

图 3.2 分布式基于主机的入侵检测结构

如图 3.2 所示，是分布式基于主机的入侵检测结构，所谓分布就是目标机拥有自己的检测引擎，能够在本地对审计数据进行检测及响应，而控制台只是存储报警、产生报表和进行数据辨析而已。具体流程见图。

根据分布式结构的特点，我们同样可以总结出它的优缺点：

优点在于：一，每个操作都是实时进行的，包括告警与响应。二、基本不影响网络通信量，由于在网络中传输的紧紧只是告警数据。

缺点在于：一，由于检测引擎在目标机上，所以会降低目标机的性能。二，不能实现多主机联合检测，因为检测的只是单一的目标机。三，没有完整的原始数据档案，必要时不能用来支持起诉。四，降低了数据辨析的能力，因为记录中只有告警数据。

3.2.3.2 基于主机的审计数据源

数据源是任何基于主机的入侵检测系统的心脏。系统的好坏取决于数据是否处理得好。数据源包括操作系统日志、应用程序日志及中间件日志。

通常一个正确放置的应用程序事件能代替几千个操作系统事件，但是大多数基于主机的入侵检测系统仍主要使用操作系统日志而不是应用程序日志，这是因为大多数操作系统都能产生事件日志，而应用程序就不一定能产生事件日志。选

择适当的审计源是主机入侵检测系统需求分析的一个关键阶段。

1、操作系统事件日志

操作系统事件日志具有多种类型、品质。好的事件日志包含许多信息，而且被保护在操作系统内核中。这样的保护就意味着它们不容易被修改、损害或哄骗。

(1) UNIX Syslog。大多数 UNIX 类操作系统都提供了 Syslog 作为一般的记录工具。Syslog 具有很不严格的 ASCII 格式，任何应用程序都可以对系统日志进行写操作。但因为 Syslog 并不是作为 UNIX 内核的一部分被保护的，所以容易被修改、哄骗。

(2) UNIX 二进制内核日志。内核日志是由 UNIX 内核执行写操作的，它在大多数 UNIX 的实现中是最靠近 TCB(置信计算库)的。它们通常反映了复杂的内核行为，而这恰恰是需要为进行有效入侵检测而记录的活动。因为数据的内容更多、更可信，所以内核日志比系统日志更好。

(3) Windows NT/2000 安全事件日志。Windows NT 及 Windows 2000 提供了置信安全日志。它有 52 种事件类型，并处于严密控制之下。Windows NT/2000 中的一个强有力的特征是能基于每个对象进行审计。这个粒度级别在创建高效的审计策略中是必不可少的。而大多数 UNIX 类操作系统却没有这项功能。

(4) 其他操作系统。大多数操作系统提供一些内核级或系统日志级审计。但 Linux 及 Window 95/98 没有事件日志。缺乏一个健壮的审计子系统的操作系统通常在安全的其他许多方面也存在缺陷。

2、中间件应用程序审计源

中间件应用程序能提供某些信息，这些信息描述的是发生在中间件中的大量事务。中间件审计源可能是比操作系统日志文件更好的审计源，因为它将许多使用该中间件的应用程序联系在一起，而操作系统日志在事务、用户及应用程序之间没有固有的相关性。

(1) 关系型数据库。RDBMS(关系型数据库管理系统)中间件应用程序(如 Oracle 及 Sybase)提供关于访问敏感数据的表、行、列的详细数据。数据库审计源颇为有用，因为许多定制应用程序要利用数据库。这就允许监控这些本身没有审计源的定制应用程序。数据库审计源通常作为一个受保护表保存在数据库本身。

3、应用程序审计源

应用程序审计源通常最有用，因为它们具有相当固有的相关特性，即具有事

务、用户、应用程序之间的相关性。

(1) 防火墙。因为与网络有关，所以防火墙审计源通常被误认为是用于网络入侵检测的。但防火墙审计源实际上是应用程序审计源的一个特殊例子。该审计源中的事件通过网络上的瓶颈能反映访问控制情况。防火墙源通常是基于 ASCII 的，并通过应用程序界面进行配置。事件通常反映所使用的服务、通信量的方向、源地址、目的地址。

(2) Microsoft Exchange/UNIX Sendmail。Exchange 及 Sendmail 是消息管理应用程序，这些程序通常是公司基础设施的一个关键部分。Exchange 日志及 Sendmail 日志中的事件既能反映常规的消息事务，也能反映功能的失效。

3.2.3.3 基于主机的入侵检测的好处

基于主机的入侵检测的好处包括检测威胁、响应、威慑、攻击预测及毁坏情况评估。如果审计数据来自置信源而且其完整性得以保护，那么起诉支持也是可能的。

(1) 威慑内部人员。基于主机的系统具有强大的威慑作用，与摄像机的威慑作用很相似。

(2) 检测。基于主机的入侵检测系统检测的活动范围很广，能确定某些威胁，也能作为判定支持系统。

(3) 通告及响应。大多数入侵检测系统在检测到入侵后能进行响应。这些响应可能是自动或人工操作，包括本地及远程操作、通告。

(4) 毁坏情况评估。入侵检测系统能维护一个极好的信息档案，用数据辨析工具可以从该档案中提取有用信息，进行毁坏情况评估。

(5) 攻击预测。在导致真正的损失之前，许多攻击可以由一些初始活动来描述其特征，入侵检测系统可以检测初始活动来预测攻击。

(6) 诉讼支持。基于主机的工具能提供一些信息来支持对计算机犯罪的起诉。这些数据包括带有特定日期、时间的文件及计算机访问模式。

3.2.4 入侵检测系统的发展趋势及存在的问题[35][33]

3.2.4.1 入侵检测系统的发展趋势

一个好的入侵检测系统应该准确，且具有好的适应性和可扩充性。今后的入侵检测系统将会更加注重用户的需求，并向下面几个主要的方向发展：

(1) 信号处理

信号处理是对付商用 IDS 显示大量误报警信息的最新且很有前途的一种方法。让操作员一直坐在 IDS 的监控台前，还要有效地监视入侵是非常乏味的事情，也不可能。所以，改进的 IDS 要有办法通过处理信号噪音的方式过滤掉那些被认为是不重要的警报。

(2) 数据挖掘

目前数据挖掘在学术界和实业界得到了广泛的重视，微软总裁比尔·盖茨预计数据挖掘技术将是今后计算机技术发展的第二方向。在安全专家界这也是种很时髦的想法。如果能对众多的原始数据进行很好地管理和分析，将可以得出很多潜在的有用信息。而数据挖掘技术解决的主要问题就是如何从大量数据中提取对用户有用的信息。

3.2.4.2 目前入侵检测系统存在的问题

评价一个入侵检测系统的好坏，一般从三个方面来考虑，即入侵检测系统的有效性(effectiveness),适应性(adaptability)和可扩展性(extensibility)。有效性是指入侵检测系统具有高的检测率和低的误报率；适应性是指入侵检测系统不仅可以检测已知的各种攻击，还能检测到已知攻击方法的变种，或者其结构可以很快适应新的攻击手法；可扩展性是指入侵检测系统易于与新的检测模型合并，或者易于根据不同的网络系统环境作出相应的定制。

目前的入侵检测系统存在以下问题：

(1) 缺乏有效性

当前大部分入侵检测产品中检测入侵的规则和模式以及统计的特征往往是专家根据经验编写的。然而，就目前复杂的网络状况，单凭专家的经验是不完整、不精确的。

(2) 缺乏适应性

编写检测代码时，专家一般着重分析目前已知的各种攻击手法和系统漏洞，导致入侵检测系统可能无法检测将来出现的未知的攻击。此外，由于其学习方法的局限性，更新速度慢，使现有的入侵检测系统难以适应目前层出不穷的新的攻击手法和各种系统漏洞。

(3) 可扩展性有限

由于入侵检测系统是专家根据经验设计的检测模型，这种模型具有一定针对

性，只适合特定的网络环境。在新的网络环境中，原有的检测规则和检测模型一般很难修改或者与新的检测模型合并。

综上所述，针对目前网络环境复杂多变、用户及系统产生大量审计数据的情况，我们需要一种有效性高、适应性强、扩展性好的入侵检测系统，对规则库能进行自动更新与维护，这就引出了数据挖掘的概念。下节将介绍数据挖掘技术。

3.3 数据挖掘技术

3.3.1 数据挖掘[26]概述

3.3.1.1 数据挖掘的诞生[27]

随着人们认识和管理水平的提高，对客观世界的描述愈来愈全面，存储的数据量愈来愈大。然而，对数据库中数据的开发应用主要是检索查询，效率很低，而且相当数量的数据具有很强的时效性，数据的价值随着时间的推移而迅速降低。简单的数据查询或统计虽然可以满足某些低层次的需要，但人们更为需要的是从大量数据资源中挖掘出有指导意义的一般知识，这些知识是对大量数据的高度概括和抽象。对于快速增长的海量数据，如果没有强有力的工具来帮助进行知识发现，那么大量的数据就被浪费而不能被充分利用，也就是所说的"数据丰富，信息贫乏"。其结果是重要的决策和判定不是基于数据库中丰富的信息，而是基于主观的直觉。为此，迫切需要能从海量数据库中提取有价值知识的工具，数据挖掘技术正是为满足上述要求而产生的。

3.3.1.2 数据挖掘的概念[42]

数据挖掘[5]是数据库技术、人工智能、机器学习和统计学等学科相结合的产物。简单地说，数据挖掘是从大量数据中提取或挖掘知识。一种比较公认的定义是：数据挖掘是指从数据库的大量数据中揭示出隐含的、先前未知的、潜在有用的信息的非平凡过程。

许多人把数据挖掘与 KDD 视为同义词，而另一些人则把数据挖掘作为 KDD 的一个基本步骤[41]。实际上，如图 3.3 所示，从数据库中发现知识的过程可以分为以下几个步骤[5][11][12][25]：

(1) 数据清理：消除噪声或不一致数据。

(2) 数据集成：将多种数据库中的数据组合在一起。

(3) 数据选择：从数据库中检索与分析任务相关的数据。

(4) 数据变换：将数据变换或统一成适合挖掘的形式。

(5) 数据挖掘：它是基本步骤，使用智能方法提取数据模式。

(6) 模式评价：根据某种兴趣度度量，识别表示知识的模式。

(7) 知识表示：使用可视化和知识表示技术，向用户提供挖掘的知识。

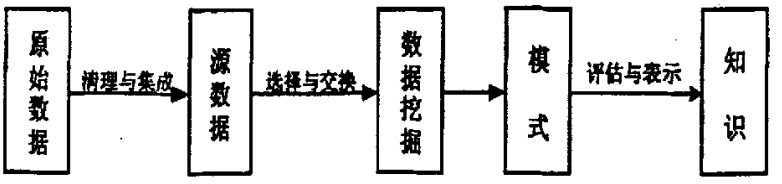

图 3.3 数据库中知识发现的过程

基于广义的观点，数据挖掘是指从存放在数据库、数据仓库或其它信息库中的大量数据中挖掘知识的过程。

3.3.1.3 数据挖掘系统

典型的数据挖掘系统[7]具有以下主要成分，如图 3.4 所示。

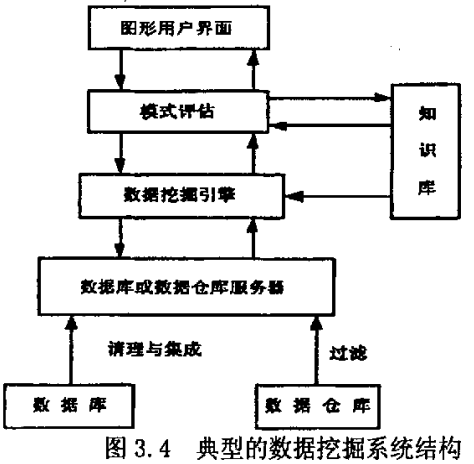

图 3.4 典型的数据挖掘系统结构

(1) 数据库、数据仓库或其它信息库。这是一个或一组数据库、数据仓库、电子表格或其它类型的信息库，可以对其进行数据清理和集成。

(2) 数据库或数据仓库服务器。根据用户的数据挖掘请求，服务器负责提取相关数据。

(3) 数据挖掘引擎。它是数据挖掘系统的基本部分，由一组功能模块组成，用于特征化、关联、序列、分类、聚类分析以及演变和偏差分析。

(4) 模式评估。通常，它使用兴趣度度量，并与数据挖掘模块交互，以便将

搜索聚焦在模式上。

(5) 知识库。存放领域知识用于指导搜索，或评估结果模式的兴趣度。

(6) 图形用户界面。该模块在用户和数据挖掘系统之间通信，允许用户与系统交互，指定数据挖掘查询任务，提供提示信息，帮助搜索聚焦。

3.3.2 数据挖掘模式[39]

数据挖掘的任务是从数据集合中发现模式。模式是用文字叙述的表达式，它可以用来描述数据集中数据的特性，表达式所描述的数据通常是数据集合的一个子集。

表 3.2 数据挖掘的分类

分类的角度	按功能分类		按算法分类			
类 型	预测型模式	描述型模式	关联模式	序列模式	分类模式	聚类模式

1、按功能分类

按功能分类数据挖掘模式可分为两大类：

(1) 预测型模式

预测型模式是可以根据数据项的值精确地确定某种结果的模式。挖掘预测型模式所使用的数据也都是可以明确知道结果的。例如，根据某种动物的资料，可以建立这样的模式：凡是胎生的动物都是哺乳类动物。当有新的动物资料时，就可以根据这个模式判别此动物是否是哺乳动物。

(2) 描述型模式

描述型模式是对数据中存在的规则做一种描述，或者根据数据的相似性把数据分组。描述型数据不能用于预测。例如，在地球上，70%的表面被水覆盖，30%是土地。

1、按算法分类（具体算法将在下一章介绍）

按算法分类数据挖掘模式可分为以下四类：

(1) 关联模式

关联模式是数据项之间关联规则，它寻找在同一个事件中出现的不同项之间的相关性。关联规则可记为 A→B。A 称为前提或者左部（LHS）， B 称为后续

或右部（RHS）。如果 A 表示“买牛奶”，B 表示“买面包”，上面的规则就是说“买牛奶的人也会买面包”。

(2) 序列模式

序列模式与关联模式相仿，它把数据之间的关联性与时间联系起来，寻找的是事件之间在事件上的相关性。为了发现序列模式，不仅需要知道事件是否发生，还要知道事件发生的时间。如在购买彩电的人们当中，60%的人会在 3 个月内购买影碟机。

(3) 分类模式

分类要解决的问题是为一个事件或对象归类。分类模式实际上是一个分类函数，能够把数据集中的数据项映射到某个给定的类上。此模式既可以分析已有的数据，也可以预测未来的数据。分类算法的工作方法是通过分析已知分类信息的历史数据总结出一个预测模型，通过预测模型来分类未知数据。用于建立分类模型的数据称为训练集，通常是已经掌握的历史数据，也可以是通过实际经验得到的数据。

(4) 聚类模式

聚类模式是把数据划分到不同的组中。组之间差别尽可能大，组内的差别尽可能小。与分类模式不同，进行聚类前并不知道将来要划分成几个组和什么样的组，也不知道根据哪些数据项来定义组。因此在聚类之后要有一个对业务很熟悉的人来解释这些组的含义。如果产生的模式无法理解或不可用，则该模式可能没有意义，需要回到上个阶段重新组织数据。

综上所述，我们知道不同的模式有不同的特点，在解决实际问题时，需要根据具体情况来结合使用多种模式。

3.3.3 数据挖掘的应用和前景

数据挖掘可以应用在各个不同的领域，举例说明：享有盛誉的市场研究公司，如美国的 A.C.Nielson 和 Information Resource，欧洲的 GFK 和 Infratest Burk 等纷纷开始使用数据挖掘工具来应付迅速增长的销售和市场信息数据；英国广播公司（BBC）也应用数据挖掘技术来预测电视收视率，以便合理安排电视节目时刻表；哥伦比亚大学由 Sal Stolfo 教授带领的团队成功地将一个基于代理的分布式数据

挖掘框架应用于检测信用卡欺骗的问题上；我国的公安部门也在研究如何利用数据挖掘技术来总结各类案件的共性和发生规律，从而在宏观上制定最有效的社会治安综合治理的方案和措施，在微观上找出犯罪人的特点，划定罪犯的范围，为侦破工作提供方向。

目前，数据挖掘在学术界和实业界已得到了广泛的重视，一份最近的 Garnter 报告指出了五项在今后 3-5 年内对工业将产生重要影响的关键技术，其中数据挖掘和人工智能排名第一。美国政府已经斥资几十亿美元用于数据挖掘技术的研究。国内一些科研单位和大学也已经开始研究数据挖掘，并把数据挖掘用于地理信息系统、遥感分析、入侵检测、故障诊断等方面，但形成的产品还比较少，此外，用数据挖掘进行企业决策的企事业单位在国内也渐渐出现，并有日益增多的趋势。

3.4 基于数据挖掘的入侵检测技术的研究现状

数据挖掘是一种特定应用的数据分析过程，可以从包含大量冗余信息的数据中提取出尽可能多的隐藏知识，从而为作出正确的判断提供基础。将数据挖掘技术应用到入侵检测中，用于对海量的安全审计数据进行智能化处理，目的是抽象出利于进行判断和比较的特征模型，这种特征模型可以是基于误用检测的特征向量模型，也可以是基于异常检测的行为描述模型。

目前，对数据挖掘算法的研究已经比较成熟，有许多算法可以使用。然而，真正要从海量数据中提取出我们所感兴趣的数据信息(知识)，需要强调的一点是“特定应用”。算法实现必须建立在特定应用的基础之上，并且需要具有足够的先验知识。

基于数据挖掘的入侵检测技术将入侵检测看作是一种数据分析过程，运用适当的数据挖掘算法对海量的安全审计数据进行自动和系统地挖掘，以求最终建立一套自适应的、具备良好扩展性的入侵检测系统。

根据调查结果，将数据挖掘应用于入侵检测已经成为一个研究热点，在这个领域已经有了近百篇论文。但是真正实现这样一套系统的还不多见，主要是 Columbin University 的 Wenke Lee 研究组和 University of New Mexico(UNM)的 Stephanie Forrest 研究组。国内这方面的研究则刚刚起步，中国科学院的国家信

息安全重点实验室、东北大学国家软件工程研究中心等走在前列。

1、Wenke Lee 研究组[13][14][15][16]

Columbia University 的 Wenke Lee 研究组在 1998 年参加了由美国国防部高级研究计划署(DARPA)资助的 Intrusion Detection Evaluation 计划，测试由 MIT 的 Lincoln 实验室提供的模拟军事网络环境中所记录的 7 周的网络流量和主机系统调用记录日志，这些数据全部采用 tcpdump 和 Solaris BSM Audit Data 的格式，包括了大约 500 万次会话，其中包含上百种攻击。这些攻击分为下面 4 种主要类型：

- 拒绝服务攻击(DOS)，如 Ping of Death、TearDrop、Smurf、SYN Flood 等等；
- 远程攻击(R2L)，如基于字典的口令猜测；
- 本地用户非法提升权限的攻击(U2R)，如各种各样的缓冲区溢出攻击；
- 扫描(PROBING)，包括端口扫描和漏洞扫描。

Wenke Lee 研究组分别从网络和主机两方面进行了审计数据的挖掘处理。针对网络数据，Wenke Lee 的主要做法是使用网络服务端口(Service)作为网络连接记录的类型标识，根据大量的正常记录生成各个服务类型的分类模型，在测试过程中，根据分类模型对当前的连接记录进行分类，并与实际服务类型进行比较，从而判断出该分类模型的准确性。针对主机数据，Wenke Lee 则使用了一种快速的规则学习算法 RIPPER，通过对正常调用序列的学习来预测随后发生的系统调用序列，并对结果进行了进一步的抽象分析，以降低算法的预测误差。根据 DARPA 报告，由 Columbia University 实现的基于数据挖掘的入侵检测系统在检测拒绝服务攻击和扫描方面优于其它系统，在检测本地用户非法提升权限方面与其它系统大致持平，在检测远程攻击方面，所有的系统表现都不令人满意，检测率都在 70%以下。

2、Stephanie Forrest 研究组[1][4][6][10]

University of New Mexico(UNM)的 Stephanie Forrest 研究组进行的是针对主机系统调用的审计数据分析处理，最初的思想是基于生物免疫系统的概念。无论是针对生物机体还是针对计算机系统，免疫系统的关键问题在于：使用一组稳定的、并且在不同个体之间存在足够差异的特征(features)来描述自我，从而使系统具备判断“自我/非自我”的能力。

然而，对于计算机系统来说，要解决这个问题相当困难。第一，恶意代码隐藏在正常代码之中难以区分；第二，系统可能的状态几乎是无限的，寻找一组稳定的特征来定义自我并不容易。Stephanie Forrest 使用短序列匹配算法对特定的特权程序所产生的系统调用序列进行了细致的分析，在这一领域作出了大量开创性工作。

在这之后，UNM 的另一个研究小组使用了有限自动机(FSM)来构建系统调用的描述语言，但是这种方法的效率和实用性都很差。Lowa State University 的一个小组实现了一种描述语言 Auditing Specification Language(ASL)，以描述程序的正常行为。另外，还有其它一些研究者采用了神经网络等人工智能的办法。

大量实验和测试结果表明，将数据挖掘技术应用于入侵检测在理论上是可行的，在技术上也是可能的。其技术难点主要在于如何根据具体应用的要求，从我们关于安全的先验知识出发，提取出可以有效地反映系统特性的特殊属性，应用合适的算法进行挖掘。技术难点还在于结果的可视化以及如何将挖掘结果自动地应用到实际的入侵检测系统中。

本文将在第四章具体叙述基于数据挖掘的主机入侵检测技术在我校一卡通系统网络安全中的具体应用。

3.5 本章小结

本章详细介绍了本文所采用的相关技术，从对网络安全的简要介绍开始，依次介绍了入侵检测技术和数据挖掘技术，并说明了基于数据挖掘的入侵检测技术的研究现状。为论文的进一步深入提供了理论基础。

第四章　系统详细设计[18-19][21][23][29-32][34][36-37]

本章是本文的核心章节，详细论述了基于数据挖掘的主机入侵检测系统的设计方案，并对方案中各模块的工作原理进行了叙述，最后提出了一种支持度递减滑移窗递增的层次挖掘算法作为本系统中规则挖掘的核心算法，用来对标准审计数据进行挖掘，实现规则库的动态维护。

4.1 概述

将数据挖掘技术应用于入侵检测系统中，是目前入侵检测领域一个比较重要的方向。数据挖掘技术非常适用于处理海量数据，从中抽象出利于进行判断和比较的特征模型，这种特征模型可以是基于误用检测的特征向量模型，也可以是基于异常检测的行为描述模型。

考虑到目前南昌大学一卡通网络的现状：专网搭建，一方面由银行前置机通过路由器连接建行网络，银行前置机负责银行方面对一卡通数据库服务器的操作，另一方面由一卡通 Web 服务器连接下层多个管理工作站，Web 服务器负责网络内部各个管理工作站对一卡通数据库服务器的操作。由于银行前置机运行的是自动程序，用来检查系统内余额不足的账户并及时进行圈存，不涉及人为的操作，所以人为攻击的可能性很小，而 Web 服务器上连数据库服务器，下连多个管理工作站，位于整个系统的关键位置，并且涉及的基本上都是人为操作，攻击可能性相对较大，所以本文设计的是一卡通网络内部 Web 服务器上的入侵检测系统。

虽然基于主机的入侵检测系统本身需要占用服务器的计算和存储资源，但是出于全面和深入分析的需要，有时可能出现巨大的数据运算量，如对一段时期内的历史数据进行综合分析，而这种分析由于会占据大量的系统资源，我们不得不在非实时的离线情况下完成。同时，为了尽量提高系统对入侵的反映速度，实时的在线入侵检测又是极其重要的。此外，由于网络攻击包括已知攻击和未知攻击两种，其中未知攻击又分为已知攻击的变种和全新的攻击。考虑到以上几方面，本文采取“在线离线结合，误用异常互助，分类聚类兼顾”的方法，即对于已知攻

击和已知攻击的变种，采用有较高概化性的基于分类算法的误用在线入侵检测；对于全新的攻击，采用基于聚类算法的异常离线入侵检测。下图 4.1 是系统的结构设计图。

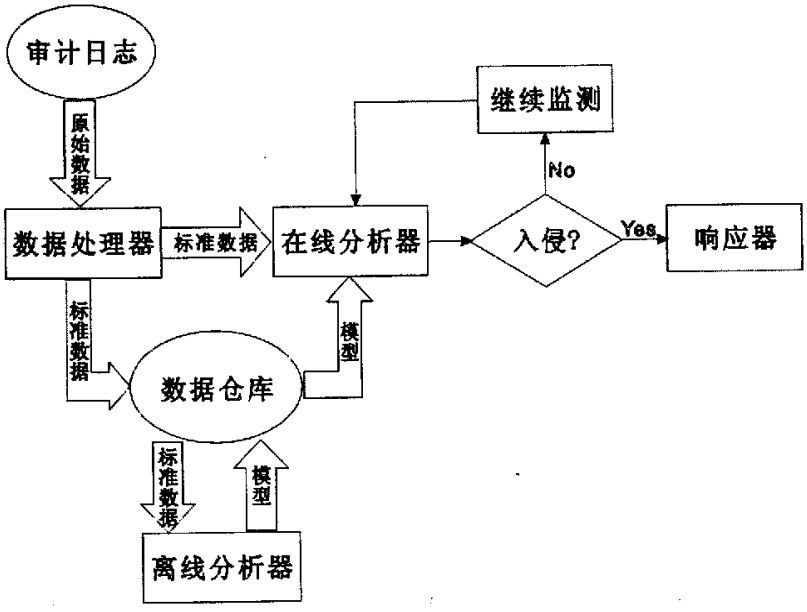

图 4.1　系统结构图

从图中可以看出，整个入侵检测系统包括五个组件：数据处理器、数据仓库、在线分析器、离线分析器和响应器。

(1) 数据处理器：负责采集原始审计数据，并预处理成标准分析数据；

(2) 在线分析器：也可以称为前台分析器，采用基于分类算法的误用检测，对实时数据进行简单迅速的模式匹配，并最终将数据实时分类为正常数据和异常数据；

(3) 数据仓库：负责存储所有已挖掘出的模型，并收集当前数据形成历史数据库，为离线分析器提供足够的分析数据源；

(4) 离线分析器：也可以称为后台分析器，采用基于聚类算法的异常检测，定期对数据仓库里的历史数据进行复杂全面的分析，以求修正旧攻击模式发现新攻击模式，并将其添加入数据仓库，再由数据仓库导入到在线分析器，使得在线检测更加全面准确；

(5) 响应器：根据在线分析器检测出的不同异常来决定具体采取何种响应策略。

4.2 审计数据的获取

审计数据的获取是主机入侵检测技术的重要基石，是进行主机入侵检测的信息来源。审计数据的获取质量和数量，决定了主机入侵检测工作的有效程度[20]。审计数据的获取工作主要需要考虑下列问题：

- 确定审计数据的来源和类型；
- 审计数据的预处理工作，其中包括记录标准格式的设计、过滤和映射操作等；
- 审计数据的获取方式，包括审计数据获取模块的结构设计和传输协议等。

下面从上述三方面结合实际情况来叙述本系统的审计数据获取。

4.2.1 审计数据类型与来源

根据目标系统的不同类型和主机入侵检测的不同要求，所需要收集的审计数据的类型不尽相同。

首先，从目标主机的类型来看，不同操作系统的审计机制设计存在差异，主机活动的审计范围和类型也有不同。典型地，UNIX 操作系统（包括 Linux、Solaris 等）与 Windows 操作系统就有较大区别，其中 UNIX 系列操作系统不同类型之间的审计系统也存在若干差异。

其次，根据不同主机入侵检测系统的设计要求和需要，其具体选取的审计数据类型和来源也各有侧重。

结合实际，一卡通系统中 Web 服务器配置的是 Windows 2000 server 系统。通过长期对 Windows 操作系统的应用，我们知道在 Windows 系统的系统工具中有多项功能可以作为主机审计数据的来源，包括：事件查看器、性能日志和警报等。

4.2.1.1 事件查看器

运行任何 Windows 2000 版本的计算机都有“应用程序日志”、“安全性日志”和“系统日志”记录事件，利用这些事件可以收集关于硬件、软件和系统问题的信息。其中应用程序日志包含由应用程序或系统程序记录的事件，例如，数据库程

序可在应用日志中记录文件错误；安全日志记录诸如有效和无效的登录尝试等事件，以及记录与资源使用相关的事件，如创建、打开或删除文件或其他对象；系统日志包含 Windows 2000 的系统组件记录的事件，例如，在启动过程将加载的驱动程序或其他系统组件的失败记录在系统日志中。

运行 Windows 并配置为域控制器的计算机有两个额外的“目录服务日志”和“文件复制服务日志”记录事件，其中目录服务日志包含 Windows 目录服务记录的事件，例如，在该目录服务日志里记录服务器和全局编录间的连接问题；文件复制服务日志包含 Windows 文件复制服务记录的事件，例如，在文件复制日志里记录文件复制失败和域控制器因为 sysvol 更改而更新时发生的事件。

运行 Windows 并被配置为域名系统 (DNS) 服务器的计算机有额外的“DNS 服务器日志”记录事件，DNS 服务器日志包含 Windows DNS 服务记录的事件。在该日志里记录有关将 DNS 名称解析成 Internet 协议 (IP) 地址的事件。

当启动 Windows 时，“事件日志”服务自动启动。所有的用户都可查看应用程序日志和系统日志。只有管理员才能访问安全日志。在默认情况下，安全日志是关闭的，要使用组策略来启用安全日志。

通过上述对事件查看器的介绍，我们可以看出 Windows 自带的事件查看器实际上可以为基于主机的入侵检测系统提供丰富的审计数据。如安全日志可以提供系统访问类型的数据，目录服务日志和文件复制服务日志可以提供目录和文件访问类型的数据，DNS 服务器日志可以提供网络访问类型的数据等等，前提是要根据不同入侵检测系统的审计需求启动相应的日志服务类型。通过将这些不同类型的日志文件导出为 Microsoft Excel 格式文件再导入到数据仓库中进行合适的数据预处理便可以为入侵检测提供丰富的审计数据。

4.2.1.2 性能日志和警报

性能日志和警报是 Windows 用来监视计算机中资源使用情况的工具。使用性能日志和警报可以自动从本地或远程计算机收集性能数据。可以使用系统监视器查看记录的计算机数据，也可以将数据导出到电子表格程序或数据库进行分析并生成报告。性能日志和警报提供下列功能：

(1) 性能日志和警报以逗号分隔或制表符分隔的格式收集数据（包括内存、处理器、磁盘和网络等），以便容易导入电子表格程序，还可以按 SQL 数据库格式收集数据。

(2) 由性能日志和警报收集的计数器数据在收集期间以及收集结束后均可以查看。因为日志作为服务运行，所以无论用户是否登录到受监视的计算机，均会进行数据收集。

(3) 可以为自动生成日志定义开始和结束时间、文件名、文件大小和其他参数。可以从一个控制台窗口管理多个日志会话。

(4) 可以设置针对计数器的性能警报，以指定当所选计数器的值超过或低于指定的设置时，便发送消息、运行程序、在应用程序事件日志中记录某项或者启动日志。

(5) 创建跟踪日志。使用默认的 Windows 2000 系统数据提供程序或另一个应用程序提供程序，当发生某些活动时（例如磁盘输入/输出 (I/O) 操作或者页错误），跟踪日志将记录下详细的系统应用程序事件。

通过上述对性能日志和警报的介绍，可以看出 Windows 自带的性能日志和警报可以收集和查看大量有关计算机中硬件资源（如内存、CPU 等）使用和系统服务活动的数据。同样通过将其导出为 Microsoft Excel 格式文件再导入到数据仓库中进行合适的数据预处理便可以为入侵检测提供有关资源消耗类型的审计数据。

本文所设计的入侵检测系统的审计数据来源正是上面所提到的一卡通系统 Web 服务器上 Windows 2000 系统中事件查看器里记录的大量的应用程序日志、系统日志和安全日志。

4.2.2 审计数据的预处理

在确定审计数据的类型和来源后，主机入侵检测所要进行的主要工作就是审计数据的预处理工作，包括映射、过滤和格式转换等操作。

预处理工作的必要性体现在以下几个方面：

1、不同目标系统环境的审计记录格式是各不相同的，对其进行格式转换的预处理操作形成标准记录格式后，将有利于系统在不同目标平台系统之间的移植；同时，有利于形成单一格式的标准审计记录流，便于后继的处理模块进行检测工作。

2、对于审计系统而言，系统中所发生的所有可审计活动都会生成对应的审

计记录，因此对某个时间段而言，审计记录的生成速度是非常快的，而其中往往大量充斥着对于入侵检测而言无用的事件记录。所以，需要对审计记录流进行必要的映射和过滤等操作。

结合实际，本系统所要处理的审计数据是事件查看器提供的应用程序日志、系统日志和安全日志。下面是对这三种日志导出为 Excel 文件格式后未进行预处理前的情况：

表 4.1　系统日志原始数据

日期	时间	来源	类型	分类	事件	用户	计算机	事件说明
1/4/2006	2:43:13 PM	RSVP	警告	无	1004 7	N/A	NC-P6QYAH FHV96P	QoS RSVP 找不到启用了通讯控制的任何界面。请通过网络和拨号连接安装 QoS 通讯控制服务。
12/30/200 5	1:03:47 PM	Tcpip	信息	无	4201	N/A	NC-P6QYAH FHV96P	系统检测到网卡 Intel(R) PRO/100 VE Network Connection 与网络连接，而且已通过网卡初始化一般操作。
12/25/200 5	2:21:25 PM	TermServ Devices	错误	无	1106	N/A	NC-P6QYAH FHV96P	无法安装打印机。

表 4.2　安全日志原始数据

日期	时间	来源	类型	分类	事件	用户	计算机	事件说明
2006/2/23	6:46:02 PM	Security	成功审核	系统事件	515	SYS TEM	NC-P6QY AHFHV96 P	受信任的登录过程已经在本地安全机制机构注册。将信任这个登录过程来提交登录申请。
2006/2/13	6:08:15 PM	Security	成功审核	登录/注销	528	NET WO RK SER VIC E	NC-P6QY AHFHV96 P	登录成功: 用户名: LOCALSERVICE 域: NT AUTHORITY 登录 ID: (0x0,0x3E5) 登录类型: 5 登录过程: Advapi 身份验证程序包: Negotiate 工作站名: 登录 GUID: {00000000-0000-0000-0000-000000000000}
2006/2/11	6:54:40 PM	Security	失败审核	策略改动	615	LOC AL SER VIC E	NC-P6QY AHFHV96 P	IPSEC 服务: IPSec 服务获得此机器上的网络接口的完整列表失败。这可能会对此机器的安全性构成潜在威胁，因为某些网络接口可能不能通过应用 IPSec 筛选器来获得希望的保护。请运行 IPSec 监视器管理单元来进一步诊断此问题。
2006/2/12	4:27:37 PM	Security	失败审核	登录/注销	529	AN ON YM OUS LOG ON	NC-P6QY AHFHV96 P	登录失败:

表 4.3 应用程序日志原始数据

日期	时间	来源	类型	分类	事件	用户	计算机	事件说明
11/12/2005	8:14:16PM	ESENT	信息	记录/恢复	300	N/A	NC-P6QYAH FHV96P	wins (1192) 数据库引擎开始恢复步骤。
1/4/2006	2:42:34PM	MSDTC	信息	SVC	4097	N/A	NC-P6QYAH FHV96P	MS DTC 已启动。
9/13/2005	11:45:30P M	ESENT	信息	常规	100	N/A	NC-P6QYAH FHV96P	wins (1224) 数据库引擎 6.01.3940.0031 已启动。
11/24/2005	1:59:47PM	SceCli	信息	无	1704	N/A	NC-P6QYAH FHV96P	组策略对象中的安全策略被成功应用。

从以上数据可以看出，Windows 事件查看器提供的原始日志数据存在统一的格式，都包括以下十个字段：日期、时间、来源、类型、分类、事件、用户、计算机和事件说明等，省去了将不同的审计记录格式转换为统一格式这一步，因此有利于形成单一格式的标准审计记录流。

在形成标准审计记录流之前，我们需要充分了解这些由事件查看器记录的原始数据，然后再结合入侵检测系统需求来对其进行分析。

因此，很有必要在这里先简要介绍一下由事件查看器记录的原始数据中各个属性的具体意义（如表 4.4 所示）：

表 4.4　原始审计数据中各属性的意义

日期	事件发生的日期。
时间	事件发生的当地时间。
来源	记录事件的软件，它可为程序名（如“SQL Server”）、系统或大程序的组件（如驱动程序名）。例如，“Elnkii”指明了 EtherLink II 驱动程序。
类型	事件安全分类：系统和应用程序日志里的错误、信息或警告与安全日志里的成功审核或失败审核。
分类	按事件来源分类事件。该信息主要用于安全日志。例如，对于安全审核，它对应于可在组策略中启用成功或失败审核的其中一个事件类型。
事件	识别特殊事件类型的编号。详细信息的第一行一般包含事件类型的名称。例如，6005 是在启动事件日志服务时所发生事件的 ID。这类事件说明的第一行是“事件日志服务已启动。”
用户	事件发生所代表的用户的名称。
计算机	产生事件的计算机的名称。
事件说明	是对发生事件的具体说明，它的格式和内容可能会根据事件类型的不同而有所变化。

结合具体需求，通过对一卡通 Web 服务器上大量的原始日志数据的分析，发现：

(1)‘事件说明’字段和‘事件’字段是一一对应的，‘事件说明’是‘事件’的具体说明，而且考虑到标准审计记录流简单化的问题，所以应该将‘事件说明’字段从原始数据中删去，由‘事件’字段足以表明所发生的事件；

(2) 由于监视的是同一台 Web 服务器，所以‘计算机’字段只有一个常量值“NC-P6QYAHFHV96P”，因此对其进行分析没有任何意义，也应该将其删去；

(3) 原始数据中‘时间’字段是具体的某个时间，是字符型的字段，考虑到检测过程中可能通过对某个具体用户的登陆时间来判断正常或异常性，所以为了便于对‘时间’字段进行数值比较，可以将其转化为数值型的字段。以“2:42:34PM”为例，将时、分、秒分为三个字段，每个字段类型都是数值型，因此将其转化为

Hour=14, Minute=42, Second=34（其中 Hour=0~23, Minute=0~59, Second=0~59）。在文中所设计的入侵检测系统中如果将早八点到晚六点作为正常登陆时间的话，那么就可以仅根据字段 Hour 来进行判断；而考虑到在进行频繁序列模式挖掘时，可能涉及到滑移时间窗为 n 秒的情况，此时则应该结合所有 Year, Month, Date, Hour, Minute, Second 来考虑时间差；

(4) 同样的'日期'字段是具体的某一日期，也是字符型字段，考虑到一卡通网络一年之中在寒暑假期间是关闭的，所以如果通过对某个具体用户的登陆日期来判断正常或异常性的话，也应将其转化为数值型进行比较，前提是要求系统管理员对每年寒暑假的具体日期进行指定。以"1/1/2006"为例，将年、月、日分为三个字段，每个字段类型都是数值型，因此可以转化为 Year=2006, Month=1, Date=1（其中 Year=2005~2015, Month=1~12, Date=1~31），如果说 2006 年 1 月 23 日到 2006 年 2 月 20 日为寒假的话，那么可以结合 Year、Month 和 Date 三个字段来进行判断。当然以'日期'字段来判断异常的可行性较小，所以也可以省去该字段，为了全面考虑，本系统将'日期'字段转换的'年'、'月'、'日'字段归为标准审计记录的一部分；

(5) 其他'来源'、'类型'、'分类'、'事件'、'用户'字段，分别说明了一个审计记录的不同属性，已经具备数据简化性与说明性的要求，所以本系统没有对其进行进一步的处理。

虽然，这些数据来自不同种类的日志，其中的每个字段在不同的日志中的重要程度可能不一样，但是考虑到预处理器设计的便利，此处没有对其予以区分，采用的是同一种预处理方法，而将具体的区分放在规则挖掘时进行考虑，如挖掘规则时针对不同的日志选取不同的关键属性和参考属性(这将在后续章节具体叙述)。

以下是对上面三个原始日志数据进行预处理以后的结果：

表 4.5 系统日志预处理数据

Year	Month	Date	Hour	Minute	Second	Source	Type	Sort	Event	User
2006	1	4	14	43	13	RSVP	警告	无	10047	N/A
2005	12	30	13	3	47	Tcpip	信息	无	4201	N/A
2005	12	25	14	21	25	TermServDevices	错误	无	1106	N/A

表 4.6 安全日志预处理数据

Year	Month	Date	Hour	Minute	Second	Source	Type	Sort	Event	User
2006	2	23	18	46	2	Security	成功审核	系统事件	515	SYSTEM
2006	2	13	18	8	15	Security	成功审核	登录/注销	528	NETWORK SERVICE
2006	2	11	18	54	40	Security	失败审核	策略改动	615	LOCAL SERVICE
2006	2	12	16	27	37	Security	失败审核	登录/注销	529	ANONYMOUS LOGON

表 4.7 应用程序日志预处理数据

Year	Month	Date	Hour	Minute	Second	Source	Type	Sort	Event	User
2005	11	12	20	14	16	ESENT	信息	记录/恢复	300	N/A
2006	1	4	14	42	34	MSDTC	信息	SVC	4097	N/A
2005	9	13	23	45	30	ESENT	信息	常规	100	N/A
2005	11	24	13	59	47	SceCli	信息	无	1704	N/A

可以看出，通过预处理以后的标准审计数据省去了很多原始数据中的冗余字段，同时也便于运用数据挖掘算法进行规则的挖掘。

4.2.3 审计数据的获取方式

标准审计数据的获取通常是由审计数据的获取模块(即预处理模块)提供的，审计数据获取模块的主要作用是获取目标系统的审计数据，并通过预处理工作后，最终目标是为入侵检测的分析处理模块提供一条单一的审计记录块数据流，供其使用。审计数据获取模块的具体设计根据主机入侵检测系统的具体特点，而有所区别。最简单的情况是审计数据获取模块与检测分析处理模块同时留驻在目标主机系统上。文中所设计的入侵检测系统就属于该类型，审计数据获取模块只需要提供经过处理后的审计记录块数据流即可，而原始审计记录的获取由 Windows 自带的事件查看器来完成。

从原始数据转换成标准审计数据的处理算法流程简要如下：在预处理模块中有一个关键的静态状态指示变量 audit_state，其作用是跟踪当前预处理器的工作状态变化情况。当预处理器刚开始启动时，该变量取值为 START；当预处理器需要打开一个审计数据文件时，状态指示为 NEXT_FILE；当预处理器从文件中读取审计记录时，当前状态为 READ_FILE；最后，当预处理器关闭当前审计数

据文件时，状态指示为 CLOSE_FILE。（这里所讲到的审计数据文件均为事件查看器导出的 Excel 文件）

```
//Processing Procedure for Preprocessor
1    While True do begin
2      If audit_state=START then begin
3        Allocate memory for audit record
4        Allocate memory for the ptr of size BLOCK_SIZE
5        audit_state=NEXT_FILE
6      Endif
7      If audit_state=NEXT_FILE then begin
8        Open audit file
9        Obtain preceding filename
10       audit_state=READ_FILE
11     Endif
12     If audit_state=READ_FILE then begin
13       Attempt to read a block from audit file to ptr
14       If successful then begin
15         If the end of audit file then
16           audit_state=CLOSE_FILE
17         Else begin
18           Advance ptr to the first record of the block
19           Call Filter_it to filter the record
20           If record passed through the filter then
21             return(audit_record)
22           Else
23             Refilter the record
24           If not the last record
25           Continue to filter the next record
26           Endif the last record
27         End else
28         If not the last block
29         Continue read next block from audit file to ptr
30         Endif the last block
```

```
31          Else
32              Reread the block from audit file
33          Endif
34          If audit_state=CLOSE_FILE then begin
35              Obtain next audit file name
36              Close current audit file
37              audit_state=NEXT_FILE
38          Endif
39      Endwhile
```

分析上述算法可知，预处理器模块启动以后进入一个无休止的循环过程：不断读取原始审计日志文件，并对每个原始审计日志文件以一定长度（block）的记录个数为一个处理单元，分别对 block 中的每条记录进行顺序的数据过滤及映射等预处理，最终返回预处理后的数据，送交在线分析器和数据仓库。

4.3 规则挖掘

4.3.1 基本算法

4.3.1.1 关联规则

关联规则是表示数据库中同一条记录不同属性之间某种横向关联关系的规则。关联规则挖掘的对象一般是事务数据库，如在本文所设计的入侵检测系统中，存储标准审计记录的数据仓库就是一种事物数据库，通过对它进行关联规则挖掘来发现每条标准审计记录中各属性之间是否存在某种关联关系。

关联规则可以用如下数学模型描述：设 $I=\{i_1, i_2, \ldots, i_m\}$是一组项集，其中 $i_1, i_2, \ldots, i_m$ 为项。$D=\{T_1, T_2, \ldots, T_n\}$是一事务集，其中 $T_1, T_2, \ldots, T_n$ 为事务，D 中的每个事务 T 是一组项，显然满足 $T \subseteq I$。如果 $X \subseteq T$，称事务 T 支持项集 X。关联规则是如下形式的一种蕴含：$X \rightarrow Y(c, s)$ 其中 $X \subseteq I$，$Y \subseteq I$，且 $X \cap Y=\phi$，s 为支持度(support)，c 为可信度(confidence)，分别可用公式表示为：

$$\text{support}(X \rightarrow Y) = \frac{\text{D中同时支持X和Y的事务个数}}{\text{D中事务的总个数}}$$

$$\text{confidence}(X \rightarrow Y) = \frac{\text{D中同时支持X和Y的事务个数}}{\text{D中支持X的事务个数}}$$

如果支持度和可信度都超过其各自的阈值（即最小支持度和最低可信度，具体值得依靠专家知识根据具体情况而定），则意味着 $X \rightarrow Y$ 可以看成 D 中的一个关联规则，表示为：$X \rightarrow Y(c, s)$。

下面结合实际来对关联规则进行解释：I={ Year=2006,..., Month=1,..., Date=1,..., Hour=14,..., Minute=43,..., Second=13,..., Source=RSVP,..., Type=警告,..., Sort=无,..., Event=10047,..., User=N/A,...}是由标准审计数据各属性的所有可能取值组成的项集，D={{ Year=2006, Month=1, Date=1, Hour=14, Minute=43, Second=13, Source=RSVP, Type=警告, Sort=无, Event=10047, User=N/A}, { Year=2005, Month=12, Date=30, Hour=13, Minute=3, Second=47, Source=Tcpip, Type=信息, Sort=无, Event=4201, User=SYSTEM },...}是由所有单条标准审计记录组成的事务集。对事务数据库 D 进行关联规则挖掘，即挖掘形如：Source = RSVP → Event = 10047 (0.5，0.03)的关联规则，该规则的支持度为 0.03，可信度为 0.5，可以解释为：在发生来源为 RSVP 的事件中有 50%的事件是编号为 10047 的事件，而 RSVP 来源的 10047 事件占所有标准审计记录的 3%。

通过对关联规则的理解，接下来我们探讨挖掘关联规则的算法。

关联规则挖掘算法可以分为两部分：一部分是找到所有支持度大于等于最小支持度的项的集合，即频繁项集(Frequent Item sets)；另一部分是利用频繁项集通过验证其可信度去产生需要的关联规则。一般情况下，关联规则算法的研究主要集中在第一部分，即如何发现所有的频繁项集。目前，关联规则挖掘算法主要分为单层关联规则挖掘算法和多层关联规则挖掘算法。其中，单层关联规则挖掘算法主要有 AIS，SETM，Apriori，AprioriTid 等；多层关联规则挖掘算法主要有 Basic，Cumulate，EstMerge 等。人们对这些算法做了大量的研究工作并进行了评价，评价结果显示 Apriori 算法是一种性能较好的关联规则挖掘算法。下面分别对频繁项集挖掘算法（Apriori 算法）和关联规则发现算法进行介绍：

1、频繁项集挖掘

Apriori 算法中频繁项集的挖掘通常分为以下几个步骤进行：

(1) 频繁 1-项集 L_1 的形成：

扫描所有项集 I 中出现的项并计算其支持度，大于等于给定支持度阈值的项

形成频繁 1-项集 L_1（顾名思义 L_1 中的项只含有一项）；

(2) 频繁 k-项集 L_k 的形成：

频繁 k-项集 L_k 的形成具体又分成两步：

第一步，在频繁(k-1)-项集 L_{k-1} 的基础上用频繁候选项集生成函数 apriori_gen() 产生频繁候选 k-项集 C_k，这样能保证所有 C_k 都是 L_{k-1} 的超集。函数 apriori_gen() 具体算法如下：

以频繁(k-1)-项集 L_{k-1} 为输入，返回结果为频繁候选 k-项集 C_k，

select $p.item_1$，$p.item_2$，…，$p.item_{k-1}$，$q.item_{k-1}$，insert into c ($c \in C_k$)

from p，q (p，$q \in L_{k-1}$)

where $p.item_1=q.item_1$，…，$p.item_{k-2}=q.item_{k-2}$，$p.item_{k-1}<>q.item_{k-1}$；

for all item sets $c \in C_k$ do

for all (k-1)-subsets s of c do

if ($s \notin L_{k-1}$) then delete c from C_k；

第二步，扫描事务数据库 D，对其中每个事务确定它支持 C_k 中的哪些候选，并累计支持数。扫描结束后，检查候选集 C_k，计算 c ($c \in C_k$) 的支持度，其中支持度大于最小支持度的 c 形成 L_k；

将以上两步反复进行，直到 L_k 为空时为止。具体算法如下：

L_1 = {frequent 1-itemsets}；

for ($k = 2$； $L_{k-1} <> \phi$； k++) do

{

C_k = apriori_gen (L_{k-1})；

for all transactions $T \in D$ do

{

C_T = subsets (C_k， T)；

for all candidates $c \in C_T$ do

c.count++；

}

L_k = {$c \in C_k$| c.count >= min_sup)；

}

Answer = $\cup_K L_k$；

2、规则发现

对于$\cup_K L_k$中的每一个频繁项集 L，找到其所有的非空子集 A。如果 support (L) /support (A) >= min_conf（最小可信度），则对于每一个子集 A，输出一个规则$A \rightarrow (L-A)$。规则发现算法有两个简化规则：一、如果 L 的一个子集 A 不能产生一个规则，则 A 的子集也不会产生规则；二、如果$(L-B) \rightarrow B$成立，则所有$(L-C) \rightarrow C$成立，其中 C 为 B 的非空子集。

4.3.1.2 序列规则

序列规则是表示数据库（其中的每条记录都是有时间戳的）中在一个特定长度的时间窗口内发生的不同记录之间某种纵向序列关系的规则。序列规则挖掘的对象一般也是事务数据库，如在本文所设计的入侵检测系统中，数据仓库里存储的标准审计记录就是有时间戳的记录，通过对它进行序列规则挖掘来发现在特定长度时间窗口内发生的不同标准审计记录之间是否存在某种序列关系。

序列规则可以用如下数学模型描述：设 $X=\{i_1, i_2\}$，$Y=\{i_3, i_4, i_5\}\ldots$ 是几个事件（其中 $i_1, i_2, i_3, i_4, i_5 \subseteq I$，I 是上面提到的项集），每个事件都是一组项，分别对应事物数据库 D 中的事务。这些事件必须在一个指定的最小频率 min_fr（即滑移时间窗）内同时出现。序列规则是如下形式的一种蕴含：$X \rightarrow Y(c, s, w)$，其中 s 为支持度(support)，c 为可信度(confidence)，w 为滑移时间窗(window)。

$$\text{support}(X \rightarrow Y) = \frac{\text{以时间窗w在D内滑移的总次数中同时出现X和Y的次数}}{\text{时间窗w在D内滑移的总次数}}$$

$$\text{confidence}(X \rightarrow Y) = \frac{\text{以时间窗w在D内滑移的总次数中同时出现事件X和Y的次数}}{\text{以时间窗w在D内滑移的总次数中出现事件X的次数}}$$

时间窗 w 得依靠专家知识根据具体情况而定。此外，序列有序规则还规定事件 X 与 Y 必须按顺序先后发生，序列无序规则则没有这种限制。如果支持度和可信度都超过其各自的阈值（即最小支持度和最低可信度，具体值得依靠专家知识根据具体情况而定），则意味着 $X \rightarrow Y$ 可以看成 D 中的一个序列规则，表示为：$X \rightarrow Y(c, s, w)$。

下面结合实际来对序列规则进行解释：设 X={Source=IISCTLS，Event=1}，Y={Source=IISCTILS，Event=2}为两个事件，分别对应事务数据库 D 中的事务，对事务数据库 D 进行序列规则挖掘，即挖掘形如：$(\text{Source} = \text{IISCTLS}, \text{Event} = 1) \rightarrow (\text{Source} = \text{IISCTLS}, \text{Event} = 2)$ (1，0.05，30s)

的关联规则，该规则的支持度为 0.05，可信度为 1，滑移时间窗为 30s，可以解释为：在发生事件(Source=IISCTLS，Event=1)后一定会发生事件(Source=IISCTLS，Event=2)，且时间间隔小于 30s；而以 30s 为滑移时间窗轮寻事物数据库 D 的总次数中有 5%的情况在时间窗内同时出现事件(Source=IISCTLS，Event=1)和(Source=IISCTLS，Event=2)。为便于理解，给出事件 1 和事件 2 的详细信息：事件 1 是从用户 XX 收到 IIS 的开始命令；事件 2 是从用户 XX 收到 IIS 的停止命令。从详细信息可以看出，事件 1 和事件 2 显然存在序列关系。

通过对序列规则的理解，接下来我们简略探讨挖掘序列规则的算法，实际上和关联规则挖掘算法有很多相似之处。

序列规则挖掘算法同样也可以分为两部分：一部分是在滑移时间窗口 w 内找到所有支持度大于等于最小支持度的事件集合，即频繁序列(Frequent Sequences)；另一部分是利用频繁序列通过验证其可信度去产生需要的序列规则。下面也从两方面介绍：

1、频繁序列挖掘

频繁序列挖掘通常分为以下几个步骤进行：

(1) 排序事务数据库：

将事务数据库中的事务按时间戳进行排序（此步骤为可选项，因为通常情况下事物数据库在形成过程中就是按时间顺序进行的）；

(2) 挖掘频繁项集：

按照关联规则挖掘算法找出所有的频繁项集；

(3) 筛选频繁项集，形成最大频繁项集（也可以称为频繁 1-序列）：

根据频繁项集找出所有的最大频繁项集，即频繁 1-序列 E_1（其中，最大频繁项集可以定义为：若有几个频繁项集同时出现在同一事务中时，项数最多的那个频繁项集），并对频繁 1-序列 E_1 进行整数映射，这样便于进一步发现频繁序列；

(4) 转换事务数据库：

按频繁 1-序列 E_1 所映射的整数将事务数据库进行转化，每一事务都用对应的整数代替；

(5) 挖掘频繁 k-序列 E_k：

参照频繁 k-项集的形成过程，这里我将频繁 k-序列 E_k 的形成具体也分成两

步：

第一步，以滑移时间窗 w 遍历事务数据库 D 时滑移的总次数 F 做循环，找出每一次滑移时在滑移时间窗里出现的频繁(k-1)-序列 E_{k-1} 和不属于 E_{k-1} 的频繁 1-序列 E_1，并在 E_{k-1} 和 E_1 的基础上用频繁候选序列生成函数 apriori_seq() 产生频繁候选 k-序列 Q_k，这同样能保证所有 Q_k 都是 E_{k-1} 的超集。函数 apriori_seq() 具体算法如下：

以频繁(k-1)-序列 E_{k-1} 为输入，返回结果为频繁候选 k-序列 Q_k，

for all $f \in F$ do

search in f to find subsets(E_{k-1}) and subsets(E_1)

select $e_{k-1} \in$ subsets(E_{k-1})，$e_1 \in$ subsets(E_1)，insert into q ($q \in Q_k$)

endfor

for all sequences $q \in Q_k$ do

if (q is not unique) then delete redundant q from Q_k；ensure Q_k is a distinct sequences sets

第二步，再次以滑移时间窗 w 遍历事务数据库 D，对其中每次滑移确定它支持 Q_k 中的哪些候选，并累计支持数。遍历结束后，检查候选集 Q_k，计算 q ($q \in Q_k$) 的支持度，其中支持度大于最小支持度的 q 形成 E_k；

将以上两步反复进行，直到 S_k 为空时为止。具体算法如下（按照计数方式的分类，以下属于 AprioriAll 算法，即对所有的频繁序列包括最高频繁序列和非最高频繁序列进行计数来计算支持度，最高频繁序列顾名思义不能再被别的频繁序列所包含）：

E_1 = {frequent 1-sequences}；

for (k = 2；$E_{k-1} \neq \phi$；k++) do

{

Q_k = apriori_seq (E_{k-1})；

for all glides $f \in F$ do

{

Q_f = subsets (Q_k，f)；

for all candidates $q \in Q_f$ do

q.count++；

}

$E_k = \{q \in Q_k | q.count >= min\_sup\}$;

}

Answer = $\cup_K E_k$;

2、规则发现

对于每一个频繁序列E，找到其所有的子序列A。如果support (E) /support (A) >= min_conf（最小可信度），则对于每一个子序列A，输出一个规则$A \rightarrow (L - A)$。

在本文所设计的基于主机的入侵检测系统中，为了更全面地分析标准审计数据并从中发现规则，选择采用关联规则和序列规则结合的规则发现方式。实际上序列规则中已经融合了关联规则，如序列规则中的频繁1-序列的形成过程便是关联规则中的最大频繁项集的形成过程，所以也可以简单的理解为运用序列规则对标准审计数据进行挖掘。关联规则和序列规则结合的规则挖掘方式，使得挖掘出的规则既有横向属性之间的关联又有纵向记录之间的关联，因此是较准确的规则发现方式，能够为审计数据提供丰富而准确的信息。

然而，以上我们讨论的基本算法只是单纯出于数据挖掘的角度，没有考虑到任何领域知识，也没有涉及到任何具体应用，如果直接运用可能会产生许多不相关的规则，如从安全日志里挖掘出来的(Year=2006, Source=security), (Year=2006, Source=security)→(Year=2006, Source=security)(1，1，0.5h)，就是一条没有任何意义的序列规则，因为对于安全日志来说，字段Source的所有取值都只有security一种，而Year在2006年一整年也就只有这么一个取值。因此系统在挖掘规则时如果只是单纯使用基本算法，而没有任何的限制的话，结果将是即消耗了挖掘的时间，占用了系统的资源，又使挖掘出来的规则有效率不高。为了避免这种情况的发生，下面结合本系统中标准审计数据的具体情况，引入合适的基于以上基本算法的扩展算法。

4.3.2 扩展算法

这里我们先介绍关键属性和参考属性的概念，然后引入一种规则挖掘扩展算法——支持度递减滑移窗递增的层次挖掘算法，最后以安全日志中的标准审计数据为例详细讲述本系统中的规则挖掘。

4.3.2.1 关键属性

关键属性顾名思义就是对事件说明起着关键作用的属性。

本系统的审计数据有三个来源：系统日志、安全日志和应用程序日志，在将其处理为标准审计记录时没有进行区分，采用了同一种预处理方式，因此他们中的每一条记录都拥有相同的属性。而在建立规则时，针对不同的日志来源，这些属性在其中的重要程度是不同的。比如说在系统日志和应用程序日志中属性 Source 记录事件的来源，它可为程序名、系统或大程序的组件，针对具体事件的不同而不同，对事件说明起着关键作用，所以 Source 属性在系统日志和应用程序日志中是一个关键属性；而在安全日志中事件的来源只有一种即 security，对事件说明没有实际意义，所以 Source 属性在安全日志中是一个无关属性。因此，我们必须找出对于事件说明起着重要作用的关键属性，运用关键属性来挖掘规则。我们把包含（非仅包含，可能还有参考属性等）关键属性的规则称作相关规则，而不包含关键属性的规则称作非相关规则。挖掘规则时我们应尽量挖掘相关规则，而避免非相关规则。

在规则发现算法中加入关键属性，即要求在发现的规则（通常指频繁序列规则，因为所有的频繁关联规则都可以理解为长度为一的频繁序列规则）中每个频繁项集都必须包含关键属性。

4.3.2.2 参考属性

参考属性顾名思义就是对事件说明起着参考作用的属性。

在本系统的标准审计数据中，就有一些参考属性，可以作为记录中其他关键属性的参考来一同说明事件。比如在安全日志中的 User 属性就可以作为关键属性 Event 的参考属性，因为不同用户发生的事件通常是不相关的。举例说明：有事件一{Hour=22，Event=538，User=A}和事件二{Hour=22，Event=540，User=B}，分别是由 A 和 B 两个不同 User 执行的操作，事实上应该是不相关的，但如果其出现的频率较高的话，在挖掘规则时可能会出现{（Hour=22，Event=538，User=A）→（Hour=22，Event=540，User=B）}规则，这显然是一条和事实不符的规则，原因很简单 User A 和 User B 无关。因此为了避免挖掘出这样的无关规则，应该运用参考属性在规则挖掘时加以限制。像在上例中就应该运用 User 属性作为参考属性，即在挖掘出的每个序列规则中，要求 User 属性为同一个值，然而，这个具体的值可以不在规则中出现，因为对于整个数据仓库中的标准审计数据而言，任何一个特

定的User值可能都不是频繁的。这也是我们将其称为参考属性的原因。

在规则发现算法中加入参考属性，即要求在发现的频繁序列规则中每个频繁项集都必须包含相同取值的参考属性。

4.3.3 支持度递减滑移窗递增的层次挖掘算法

通过对关联规则、序列规则、关键属性和参考属性的介绍，下面结合实际以安全日志中的标准审计数据为例，详细讲述本系统中采用的规则挖掘算法。为了保证规则库的完整性，我们引入一种扩展挖掘算法——支持度递减滑移窗递增的层次挖掘算法。

首先解释一下支持度递减和滑移窗递增的概念。由于在数据仓库中标准审计数据可能出现各种各样的情况，规则挖掘分两种情况来分析：第一，单条记录的关联规则（只涉及支持度的大小，不涉及滑移窗的大小），因为数据仓库中有些项集出现的频率高，而有些项集出现的频率低，这样为了发现低频率的关联规则，我们考虑采取支持度递减的方式，先挖掘高支持度的关联规则再依次挖掘支持度递减的关联规则，直到支持度达到最小支持度阈值；第二，多条记录的序列规则（既涉及支持度的大小，又涉及滑移窗的大小），同样由于数据仓库中序列出现的不同情况，为了使发现的序列规则具备充分的完整性，我们考虑采取支持度递减滑移窗递增的方式，分两个步骤来完成，首先，固定一个较小的起始滑移时间窗，然后以一个较高支持度为起始支持度逐渐降低支持度到最小支持度，挖掘序列规则，再次，增大滑移时间窗，再从起始支持度依次递减到最小支持度，挖掘序列规则，这样一直进行到滑移窗增到最大而支持度减到最小为止，结束整个序列规则的挖掘过程。

下表4.8是一组安全日志中的标准审计数据：

表4.8 安全日志中的部分标准审计数据

Year	Month	Date	Hour	Minute	Second	Source	Type	Sort	Event	User
2006	1	4	7	58	2	Security	成功审核	登录/注销	528	ADMINISTRATOR
2006	1	4	10	8	15	Security	成功审核	策略改动	612	SYSTEM
2006	1	4	14	54	40	Security	成功审核	特权使用	576	ADMINISTRATOR
…	…	…	…	…	…	…	…	…	…	…
2006	1	4	18	10	35	Security	成功审核	登录/注销	551	ADMINISTRATOR
2006	1	4	22	26	29	Security	失败审核	帐户登录	680	IUSR_ZJ

我们以安全日志为例运用上面所介绍的支持度递减滑移窗递增的层次挖掘算法结合关键属性和参考属性来详细说明规则挖掘的过程：

首先，我们对安全日志中标准审计数据的十一个属性（Year，Month，Date，Hour，Minute，Second，Source，Type，Sort，Event，User）进行一一分析，选择关键属性和参考属性。

1、属性 Year、Month、Date：记录事件发生的日期。正如我们将原始数据转换为标准审计数据时将日期属性分解为年、月、日三个属性的初衷，Year、Month、Date 属性只是为了计算记录之间的时间差，便于挖掘与时间窗相关的序列规则，因此没必要将其作为关键属性来考虑。因为如果数据仓库中的审计数据只有近一两年的记录的话，那么对于 Year、Month 属性甚至 Date 属性来讲，他们各个取值的支持度都应该挺高的，毕竟他们的取值很有限（Year=2005~2006，Month=1~12，Date=1~31），但是实际上对于具体规则来讲意义不大。因此，将属性 Year、Month、Date 归为次要属性。

2、属性 Hour、Minute、Second：记录事件发生的时间。正如我们将原始数据转换为标准审计数据时将时间属性分解为时、分、秒三个属性的初衷，Hour、Minute、Second 属性一方面是为了计算记录之间的时间差，便于挖掘与时间窗相关的序列规则，另一方面 Hour 属性对于说明事件发生的时间具有较好的概括性（比如说相对于一个事件发生的具体几分几秒来说，我们通常更在意的是一个事件具体发生在几点）。因此，将属性 Hour 归为关键属性，而属性 Minute 和 Second 归为次要属性。

3、属性 Source：记录事件的来源。由于是安全日志，Source 属性的取值唯一（即 Source=Security），所以没有必要在规则中出现。因此，将属性 Source 归为无关属性。

4、属性 Type：记录事件的类型。在安全日志中，Type 属性的取值有两种（即 Type=成功审核，Type=失败审核），但是它对于判断事件是正常行为还是异常行为有一定的参考价值。虽然在大多数情况下失败审核的事件是正常行为（因为基于所有入侵检测假设，在审计数据中正常行为通常占所有记录的 98%以上，而异常行为只占 2%左右），但是从概率的角度来讲，正常行为审核失败的情况相对于其审核成功的情况是很少的，而异常行为审核失败的情况相对于其审核成功的情况却很多。因此，将属性 Type 归为关键属性。

5、属性 Sort：记录事件的分类。在安全日志中，Sort 属性的取值依赖于 Event 属性的值，Sort 和 Event 是一对多的关系，即如果 Event=528、538……，那么 Sort=登录/注销，如果 Event=612、615……，那么 Sort=策略改动，等等，可见 Event 属性的值决定了 Sort 属性的值。因此，将属性 Sort 归为无关属性（前提 Event 属性为关键属性）。

6、属性 Event：记录事件的种类。对事件说明起着关键作用，能够具体说明发生了什么事件，在审计数据中用事件的 ID 表示，每个 ID 对应一种不同的事件。因此，将属性 Event 归为关键属性。

7、属性 User：记录执行事件的用户。对事件说明起着参考作用，能够具体说明事件是由谁执行的。由于不同用户进行的操作之间是不相关的，为了在进行频繁序列规则挖掘时发现同一用户先后执行的一系列操作，而避免频繁序列规则中出现不同用户这类不合理的规则。因此，将属性 User 归为参考属性。

通过上述分析，得出以下结论：在对安全日志中的标准审计数据进行规则挖掘时，应将属性 Hour、Type、Event 作为关键属性，属性 User 作为参考属性。

结合所选择的关键属性和参考属性，以下是支持度递减滑移窗递增的层次挖掘算法在安全日志中进行规则挖掘的具体流程：

支持度递减滑移窗递增的层次挖掘算法

****

Input：标准安全审计数据库 D，关键属性，参考属性，起始关联支持度 S_i，结束关联支持度 S_t，起始序列支持度 S_i'，结束序列支持度 S_t'，起始时间窗 W_i，结束时间窗 W_t，关联可信度 C，序列可信度 C'；

Output：频繁序列规则集合 A（包括频繁关联规则，即频繁 1-序列）；

Begin

(1) R=ϕ，A=ϕ；R 存放旧关键属性值和旧参考属性值，A 存放已挖掘出的频繁序列规则。

(2) 取最小关联支持度 S（缺省情况下为起始关联支持度 S_i），扫描数据库 D，找出所有满足 S 的长度为一的由关键属性值或参考属性值组成的 R 中不存在的频繁 1-项集 L_1'；

(3) 把 $R \cup L_1$'作为频繁 1-项集 L_1，运用前面提及的关联规则挖掘算法 Apriori 算法进行频繁 k-项集 L_k 的挖掘，确保 L_k 中至少有一项是 L_1'中的新关键属性值

或参考属性值，这样可以避免对旧规则的重复挖掘，最后把 L_l'并入 R 中；

实际上，k 的取值最大为 4，因为关键属性和参考属性总共 4 项，可见不可能出现 k 大于 4 的情况。

(4) 整合$\cup_K L_k$（k=1，2，3，4）为最大频繁项集集合 L，即形成频繁 1-序列 E_1；

实际上经过这一步骤以后，最大频繁项集集合 L 中的每一元素都应该是由 Hour、Type、Event 和 User 四个属性值组成，具有统一的 (Hour=...，Type=...，Event=...，User=...) 形式。

(5) 取时间窗 W（缺省情况下为起始时间窗 W_i），最小序列支持度 S'（缺省情况下为起始序列支持度 S_i'），运用序列规则挖掘算法进行频繁 k-序列 E_k 的挖掘，最后 A'=$\cup_K E_k$；

由于是序列规则挖掘，所以必须考虑在序列规则中出现的最大频繁项集的参考属性 User 的取值相同，该步骤执行完成以后，得到的规则应该形如(Hour=...，Type=...，Event=...，User=X)→(Hour=...，Type=...，Event=...，User=X)→.....

(6) A=A∪A'，减小序列支持度 S'=S'×0.9，跳转到步骤 5，直到 S'<S_t'结束，还原 S'=S_i'；

(7) 增大时间窗 W=W×2，跳转到步骤 5，直到 W>W_t 结束，还原 W=W_i；

(8) 减小关联支持度 S=S×0.9，跳转到步骤 2，直到 S<S_t 整个规则挖掘过程结束。

(9) Return A；

End

****

假设标准审计数据库中的记录足够全面充分，那么运用上述扩展算法得到的频繁序列规则也会相应比较全面。下面举两个有代表性的规则，都是由上述算法挖掘出来的频繁序列规则，一个为正常行为规则中的一种，另一个为异常行为规则中的一种。

- 正常行为规则：

(Hour=7，Type=成功审核，Event=528，User= ADMINISTRATOR)→(Hour=18，Type=成功审核，Event=551，User= ADMINISTRATOR) [1，0.1，11h]

该规则解释为：当以 11h 为时间窗在整个标准安全审计数据库 D 中滑移时，

用户 ADMINISTRATOR（系统管理员）早上 7 点执行动作 528，接着下午 6 点执行动作 551 的事件出现的次数占整个滑移次数的 10%，而且用户 ADMINISTRATOR 在早 7 点执行动作 528，11 个小时后接着执行 551 的概率为 100%。根据规则的内容并结合实际情况，可以判定这是一条合理有效的描述正常行为的频繁序列规则，描述的是用户 ADMINISTRATOR 每天早 7 点近 8 点上班时登录系统，下午 6 点下班时退出系统的行为。（事件 528：用户 XX 登录成功；事件 551： 用户 XX 要求注销）

- 异常行为规则：

(Hour=22，Type=失败审核，Event=680，User=X) (Hour=22，Type=失败审核，Event=680，User=X)→(Hour=22，Type=失败审核，Event=680，User=X) [0.8，0.002，5m]

该规则解释为：当以 5m 为时间窗在整个标准安全审计数据库 D 中滑移时，某用户 X 在晚上 10 点重复执行动作 680 失败审核 2 次以后，继续执行动作 680 被失败审核的事件出现的次数占整个滑移次数的 0.2%，而发生 2 次后继续第 3 次的情况占 80%。根据规则的内容并结合实际情况，可以判定这是一条合理有效的描述异常行为的频繁序列规则，描述的是某用户 X 在晚上 10 点反复尝试登录系统失败的行为。（事件 680：用户 X 尝试登录）

4.4 基于分类算法的误用在线入侵检测

4.4.1 分类算法

数据分类实际上就是将数据库中的对象分为几个已知类的过程。

分类的过程分两个阶段：

第一阶段，分类的内容是训练数据库（其中的记录已经拥有明确的分类标识），目标是运用训练数据库中记录的某些特征属性，找出每个类的准确描述（即分类规则）；

第二阶段，分类的内容是实际数据库（其中的记录没有分类标识），目标是运用通过训练数据库学习到的分类规则对实际数据库中的记录进行分类，即给实际数据库中的每条记录一个分类标识。

数据分类的具体实现方法有很多，概率统计、神经网络、专家系统等，都是目前数据挖掘中分类算法研究的重要方向。常用的分类算法有 RIPPER、ID3、Nearest-neighbor、C4.5、Naive Bayes、神经网络(Neural network)等。

4.4.2 分类算法在系统中的应用

在入侵检测系统中运用分类算法，无疑就是根据分类规则将数据分为'正常行为'、'异常行为'两类。

本系统中，分类算法的运用同样分为两个阶段：

第一阶段，运用分类算法对已有分类标识的标准审计训练数据库进行分类规则学习，也就是运用上一节介绍的支持度递减滑移窗递增的层次挖掘算法进行规则挖掘，找出'正常行为'类和'异常行为'类的准确描述。

运用 RIPPER 算法，正常行为和异常行为的分类规则可以表示为：

IF (Hour=7 , Type= 成 功 审 核 , Event=528 , User= ADMINISTRATOR) → (Hour=18 , Type= 成 功 审 核 , Event=551 , User= ADMINISTRATOR) [1，0.1，11h]

THEN Normal

IF ..... THEN Normal

.....（正常行为规则）

IF (Hour=22，Type=失败审核，Event=680，User=X) (Hour=22，Type=失败审核，Event=680，User=X)→(Hour=22，Type=失败审核，Event=680，User=X) [0.8，0.002，5m]

THEN Abnormal

IF ..... THEN Abnormal

.....（异常行为规则）

很显然，这是一种模式匹配的误用检测规则。

第二阶段，运用上述分类规则对刚经过预处理器后形成的标准审计数据进行实时分类，判定标准审计数据中的事件是正常行为还是异常行为，从而实现对异常行为进行实时检测的功能。

4.5 基于聚类算法的异常离线入侵检测

4.5.1 聚类算法

数据聚类实际上就是将数据库中的对象分组成为由类似的对象组成的多个未知类的过程。

聚类与分类不同，聚类时输入数据库是一组没有任何分类标识的记录，而聚类的目的是根据最大化类内相似性而最小化类间相似性的原则，合理地对记录进行分组，并最终用不同规则来描述不同的类别。

也可以这样理解聚类与分类：聚类是无指导的学习，分类是有指导的学习，两者所采用的方法相差甚远。通常聚类比分类耗时，而且聚类算法的复杂程度比分类算法要大得多。

目前聚类算法有很多种，算法的选择取决于数据的类型和聚类的应用目的。大体上，聚类的算法可以分成下列几类:

(1) 划分方法。给定一个 N 个对象或元组的数据库，运用划分方法构造数据的 K 个划分，每个划分表示一个聚簇，并且 $K \leq N$。

(2) 层次方法。运用层次方法对给定数据对象集合进行层次的分解。根据层次分解的形成方式，又可以分为凝聚的或分裂的层次方法。

(3) 基于距离的方法。该方法把对象与现有类（第一次时现有类为空）一一进行距离比较，选择最小距离，如果最小距离小于某个阈值，就把对象归为此类，否则归为一个新类。

(4) 基于网格的方法。该方法把对象空间量化为有限数目的单元，形成一个网格结构，所有的聚类操作都在网格结构(即量化空间)上进行。

(5) 基于模型的方法。该方法为每个簇假定了一个模型，寻找数据对此模型的最佳拟合。

4.5.2 聚类算法在系统中的应用

在入侵检测系统中运用聚类算法，通常是希望能够通过聚类得到更精确的数据分组，并在分组的基础上进一步挖掘规则，从而对分组进行更准确的描述。

通过对上述聚类算法的比较，并结合实施入侵检测的两个假设：第一，在审计记录中，正常数据量远远大于攻击数据量，正常数据量一般占总数据量的98%以上；第二，正常行为的记录和攻击行为的记录在属性值分布上有非常大的区别。考虑到在本系统中对记录进行编码的可行性，最终选择了基于距离的聚类算法，通过记录的编码来计算记录之间的距离，从而实现对标准审计记录的聚类。

在引出基于距离的聚类算法之前，有必要先简单介绍一下本系统中所采用的记录编码方式。

仍然以标准安全审计数据为例，前面我们已经对标准安全审计数据库中记录的各个属性进行了分析，并选择出了关键属性和参考属性，排除了无关属性，这样我们就可以将每条记录仅用关键属性 Hour、Type、Event 和参考属性 User 来表示。因此要对记录进行编码就意味着要将 Hour、Type、Event、User 四个属性用数值表示。

进一步观察 Hour、Type、Event、User 的取值情况，我们发现 Hour 属性和 Event 属性已经具有数值的形式，Hour 是事件发生的小时，取值 Hour=0~23，Event 是事件的 ID，取值 Event=1~999（注意：这里的 999 不是说明有 999 种安全事件 ID，而只是说明可能出现的最长位数）。为了使规则编码最终拥有相同的编码长度，决定属性 Hour 用两位十进制数表示（00~23），属性 Event 用三位十进制数表示（001~999）。此外 Type 属性的取值只有两种，因此可以只用一位十进制数表示（0~1，0 表示失败审核，1 表示成功审核）。对于 User 属性我们可以这样来进行编码：首先扫描一遍标准安全审计数据库，找出不同的 User 值，并记录下 User 的个数 Count，最后用$(\log_{10}(Count)+1)$取整位十进制数来表示不同的 User 值。

运用上述的编码规则，我们可以对标准安全审计数据库中的每条记录进行编码。如记录 I=（Hour=7，Type=成功审核，Event=528，User= ADMINISTRATOR）可以编码为 i=(07,1,528,001)，而记录 J=(Hour=22,Type=失败审核,Event=680,User= IUSR_ZJ)，可以编码为 j=(22,0,680,011)。(注：User= ADMINISTRATOR，和 User= IUSR_ZJ 分别是在进行数据库扫描时出现的第 1 个值和第 11 个值，所以此处编码为 001 和 011）

根据著名的欧几里得距离度量方法，将记录之间的距离定义如下：

$$d(i,\ j)=\sqrt{\left|X_{i1}-X_{j1}\right|^2+\left|X_{i2}-X_{j2}\right|^2+\cdots+\left|X_{in}-X_{jn}\right|^2}$$

其中 $i=(X_{i1},\ X_{i2},\ \ldots,\ X_{in})$和 $j=(X_{j1},\ X_{j2},\ \ldots,\ X_{jn})$为两个记录。$X_n$ 为记录的第 n 个属性。如果直接求两个记录的距离的话我们会发现，当记录的某个属性普遍有较大值时，这个属性对距离的影响就相对较大。比如，上述的 i=（07，1，528，001）和 j=（22，0，680，011），当计算 i 与 j 两个记录的距离时，属性 Event 对距离影响最大，其次是属性 User、Hour，最后才是属性 Type。

为了解决这个问题，我们引入下列平衡算法来均衡每个属性对距离的影响程度：

假设有 N 条记录，每条记录有 M 个属性，$record_i[j]$表示第 i 条记录的第 j 个属性值，其中 $i\in N$，$j\in M$。

首先，我们计算每个属性的平均值：

$$avg\_record[j]=\frac{1}{N}\sum_{i=1}^{N}record_i[j],\ (1\le j\le M)$$

根据平均值计算出每个属性值与平均值之间差值的均值：

$$std\_record[j]=\sqrt{\sum_{i=1}^{N}(record_i[j]-avg\_record[j])^2},\ (1\le j\le M)$$

最后将第 i 条记录的第 j 个属性值表示为：

$$new\_record_i[j]=\frac{\left|record_i[j]-avg\_record[j]\right|}{std\_record[j]},\ (1\le i\le N,\ 1\le j\le M)$$

在新属性值 $new\_record_i[j]$ 的基础上使用欧几里得方法度量记录之间的距离，可以尽量减小记录中各属性在距离计算时对结果影响程度的差异。

通过上述介绍我们了解了记录之间距离的计算方式，下面结合实际引入本系统中具体采用的基于距离的聚类算法：

(1) 初始化一个空聚类集 S；

(2) 对于一个已编码的标准审计记录，如果 S 为空，把这个记录定义为簇的初始记录；如果 S 不为空，计算此记录与所有簇的距离 d，找到与此记录最近的簇，并记录距离 d_{min}。

(3) 如果 d_{min} 小于阈值 w（w 的取值得依据专家知识根据具体情况而定），则把此记录归于此簇；否则，以此记录为初始记录建立一个新簇；

(4) 重复步骤2、3，直到对所有的标准审计记录计算完毕。

(5) 对所得的簇进行标记。超过一定数据量的簇标记为正常，而小于这个数据量的簇标记为攻击。

很显然这是一种基于异常的入侵检测。由聚类分析得到的正常数据和异常数据通常是比较纯的，我们把这些正常记录和异常记录用于规则学习，能够产生更准确的正常模式和异常模式的描述。

由于系统在聚类时分析的数据源是数据仓库中所有当前存积的标准审计数据集合，其中有近期的数据也有历史数据，量非常大，涵盖面也非常广，所以通常被用来进一步校准已有的模式并发现全新的模式。此外，考虑到聚类算法的复杂程度并且涉及到大量的处理数据的问题，为了避免给应用服务器带来太大的负担，系统采取了定期离线的处理方式来实施入侵检测，并挖掘规则，最后及时把规则添加入实时在线入侵检测模块中，实现对记录更准确的检测。

4.6 本章小结

本章首先对所设计的基于主机的入侵检测系统框架进行了描述，然后依据框架的结构依次介绍了框架中出现的各个模块，包括数据预处理模块、数据仓库、在线分类误用检测模块和离线聚类异常检测模块等。在规则挖掘算法上，介绍了关联算法、序列算法、关键属性、参考属性，并在此基础上提出了一种支持度递减滑移窗递增的层次挖掘算法，最后把它作为本系统中规则挖掘的核心算法，用来对数据仓库中的标准审计数据进行规则挖掘，并最终运用于分类算法和聚类算法中，实现对数据的在线与离线检测。

第五章　实验设计与讨论

本章我们选择一个攻击特例——“口令攻击”，作为实验环境，说明本系统是如何用数据挖掘算法发现攻击模式并进行入侵检测的，最后对上一章所设计的系统的性能进行了分析讨论。

5.1 口令攻击[22]

口令是网络系统的一道防线。当前的网络系统都是通过口令来验证用户身份、实施访问控制的。口令攻击是指黑客破解合法用户的口令，或避开口令验证过程，然后冒充合法用户潜入目标网络系统，夺取目标系统控制权的过程。

如果口令攻击成功黑客进入了目标网络系统，他就能够随心所欲地窃取、破坏和篡改被侵入方的信息，直至完全控制被侵入方。

5.1.1 口令攻击的主要方法

1、猜测攻击。首先使用口令猜测程序进行攻击。口令猜测程序往往根据用户定义口令的习惯猜测用户口令，像名字缩写、生日、宠物名、部门名等。在详细了解用户的社会背景之后，黑客可以列举出几百种可能的口令，并在很短的时间内就可以完成猜测攻击。

2、字典攻击。如果猜测攻击不成功，入侵者会继续扩大攻击范围，对所有英文单词进行尝试，程序将按序取出一个又一个的单词，进行一次又一次尝试，直到成功。据有的传媒报导，对于一个有 8 万个英文单词的集合来说，入侵者不到一分半钟就可试完。所以，如果用户的口令不太长或是单词、短语，那么很快就会被破译出来。

3、穷举攻击。如果字典攻击仍然不能够成功，入侵者会采取穷举攻击。一般从长度为 1 的口令开始，按长度递增进行尝试攻击。由于人们往往偏爱简单易记的口令，穷举攻击的成功率很高。如果每千分之一秒检查一个口令，那么 86% 的口令可以在一周内破译出来。

4、直接破解系统口令文件。所有的攻击都不能够奏效，入侵者会寻找目标主机的安全漏洞和薄弱环节，饲机偷走存放系统口令的文件，然后破译加密的口令，以便冒充合法用户访问这台主机。

5.1.2 口令攻击伴随的现象

口令攻击就是不断地尝试口令以求成功登陆系统，获得操作权限。通常口令攻击伴随的现象就是“三次登陆失败”。

以我校一卡通网络为例，口令攻击在应用服务器安全日志中被记载为表 5.1 所示：

表 5.1　三次失败的登陆企图

日期	时间	来源	类型	分类	事件	用户	计算机
2006/2/23	10:11:03 PM	Security	失败审核	登录/注销	529	ANONYMOUS LOGON	NC-P6QYAHFHV96P
2006/2/23	10:11:09 PM	Security	失败审核	登录/注销	529	ANONYMOUS LOGON	NC-P6QYAHFHV96P
2006/2/23	10:11:13 PM	Security	失败审核	登录/注销	529	ANONYMOUS LOGON	NC-P6QYAHFHV96P
…	…	…	…	…	…	…	…

5.2 实验设计与讨论

5.2.1 实验设计

结合上一节所说的口令猜测的攻击现象，我们运用第四章设计的系统对其进行实现。步骤如下：

1、由审计数据的获取模块获取审计数据，并进行数据预处理，形成标准审计数据，如下表 5.2 所示。

表 5.2　三次登陆失败的预处理数据

Year	Month	Date	Hour	Minute	Second	Source	Type	Sort	Event	User
2006	2	23	22	11	3	Security	失败审核	登录/注销	529	ANONYMOUS LOGON
2006	2	23	22	11	9	Security	失败审核	登录/注销	529	ANONYMOUS LOGON
2006	2	23	22	11	13	Security	失败审核	登录/注销	529	ANONYMOUS LOGON
…	…	…	…	…	…	…	…	…	…	…

2、将标准审计数据集中到数据仓库中，然后由离线分析器运用支持度递减滑移窗递增的层次数据挖掘算法遍历数据仓库中的所有标准审计数据进行规则挖掘，将属性 Hour、Type、Event 作为关键属性，属性 User 作为参考属性。

针对上述数据可以得到如下异常行为模式：

(Hour=22，Type=失败审核，Event=529，User=X) (Hour=22，Type=失败审核，Event=529，User=X)→(Hour=22，Type=失败审核，Event=529，User=X) [0.87，0.002，15s]

3、将上述异常行为模式存入数据仓库，并及时导入在线分析器，用来对后续标准审计数据进行实时入侵检测，如果匹配的话则进行报警并响应。

这样，通过将上述异常行为模式加入到在线分析器的规则库以后，在线分析器就有了检测口令攻击的能力。

5.2.2 实验讨论

由于时间和条件的限制，这里没有列举其他攻击模式。但是，从以上实验我们可以总结出以下几点：

首先，数据挖掘算法能够代替人工规则编码，减少工作量，避免人为因素，从而更全面更客观的对规则进行挖掘。

其次，系统中的离线分析器和在线分析器能够很好的相互配合，从而对入侵进行检测。离线分析器采用复杂的聚类算法对审计数据进行挖掘，模式更准确，在线分析器采用简单的分类算法对审计数据进行判断，响应更及时。

再次，支持度递减滑移窗递增的层次数据挖掘算法比较灵活，能够挖掘出不同支持度、可信度和滑移窗的所有规则，不依赖专家知识来确定特定取值，只需限定它们的取值范围就可以。

第四，数据仓库中存储了所有的标准审计记录，必要时可以提供完整的记录集以便查看或是作为凭据。

最后，威慑作用将是该基于主机的入侵检测系统的最大好处所在。原因很简单，如果人们知道他们的行为可能被监视，那么他们干坏事的可能性就小多了。

5.3 本章小结

本章通过对“口令攻击”特例的介绍与实验，解释了系统是如何用数据挖掘算法发现攻击模式的，此外对上一章所设计的系统的性能进行了分析讨论。

第六章　结束语和展望

6.1 论文总结

本文主要做了如下工作：

1、简要介绍了校园一卡通系统，包括概念、作用、应用范围与现实意义等，给论文工作的全面展开提供了铺垫；

2、全面介绍了我校一卡通网络的现状，包括网络规模、功能、软硬件系统等，并在此基础上对一卡通系统的网络安全做出了全面的分析，为论文工作的进一步深入提供了基础；

3、简要介绍了网络安全，包括网络安全的定义、目标和技术分类等，并在此基础上系统地介绍了论文所采用的各种技术，包括入侵检测技术和数据挖掘技术等，最后介绍了数据挖掘技术在入侵检测领域的研究现状，为论文的研究工作提供了理论依据和技术支持；

4、详细论述了针对南昌大学一卡通网络所设计的基于主机的入侵检测系统，并提出了一种"在线离线结合，误用异常互助，分类聚类兼顾"的思想。文中依据系统框架结构先后说明了框架中出现的各个模块及其工作原理，包括数据预处理模块、数据仓库、在线分类误用检测模块和离线聚类异常检测模块等。

5、结合实际的审计数据着重介绍了关联算法、序列算法、关键属性以及参考属性等，以此为基础提出了一种支持度递减滑移窗递增的层次挖掘算法，并把它作为系统中规则挖掘的核心算法，运用于数据仓库中的标准审计数据的规则模式挖掘。

6、以"口令攻击"为例，说明了本系统进行规则挖掘和入侵检测的全过程，并对系统的性能进行了分析与讨论。

6.2 存在的问题

由于许多主客观的原因，本文的研究工作存在一定的局限性。

主观上，由于本人在网络安全领域涉掠不深，知识水平有限，所以论文的研究工作肯定存在许多考虑不足之处；

客观上，由于在论文完成阶段本人一直在外地实习，所以文中仅仅给出了一个基于主机的入侵检测系统的设计方案和一个简单的实验设计，并没有将其真正运用到南昌大学校园一卡通系统的应用服务器上进行测试，所以难免存在很多需要进一步改进的地方。

6.3 今后工作展望

1、正如上一节所指出的问题，文中提出的基于主机的入侵检测系统的设计方案需要进一步拿到实际环境中去加以实现，测试其对审计数据的检测性能并加以改进；

2、本文所设计的入侵检测系统是针对南昌大学一卡通网络的，由于目前有限的网络规模，所以只是将数据挖掘技术运用到了基于主机的入侵检测中，提出的规则挖掘算法也只是针对 Windows 事件查看器中的审计数据而言的，至于该算法是否也适用于网络数据流或是能否直接运用于基于网络的入侵检测系统，尚需进一步分析；

3、 系统中采取的审计数据源只是局限于应用服务器上 Windows 操作系统提供的审计日志。为了能够更准确的检测攻击，今后可以考虑将数据库服务器中关系型数据库 Oracle 系统提供的中间件应用程序审计源作为审计数据的一部分，来进行规则挖掘。

4、本文所设计的入侵检测系统是集中式基于主机的入侵检测系统，数据收集模块和检测引擎模块均位于同一台应用服务器上，这是根据目前网络状况而定的，今后可以考虑将其改进为分布式基于主机的入侵检测系统，这样能够同时保护多台主机。

5、随着校园一卡通系统在各高校的日渐普及，网络安全的问题日益凸显，本文仅以南昌大学一卡通系统作为研究背景，至于所设计的入侵检测系统是否同样适用于其他高校的一卡通系统，需要具体运用具体分析。

致 谢

在论文完成之际，本人向关心爱护我的老师、同学、同事及家人致以深深的谢意。

首先，感谢导师李建民教授对作者的关心和支持，感谢他给一直以来对学生的严格要求以及为学生提供的良好的学习环境和实践机会。导师广博精深的知识和运筹帷幄的气魄是值得学生毕生学习的榜样。

感谢付爱英老师在我论文研究阶段所给予的热情帮助，付老师曾多次亲力亲为帮我传导一卡通网络的实验数据，使得作者能够顺利完成论文的研究工作。

感谢占传杰、黄传华、曾京炜、杨伟农、刘烨、万国金等老师在三年的研究生学习阶段给予我的帮助，使得我能够顺利完成研究生阶段的学习。

感谢我的师兄陈林、刘承启、方永、林振荣、张彤、胡建平等和师姐熊艳、戴琴、吴志琼等，感谢他们在学习上给我的支持与帮助，三年来，大家在一起共同学习，努力工作，彼此促进，相互提高，关系融洽，亲如兄弟姐妹，我将永远怀念这个大家庭。

感谢上海中芯国际的同事张诗华、逄志伟、郭俊华、笪慧、刘艳等人给予本人的帮助与呵护，实习期间众位大哥哥大姐姐在工作上对我细心教导，在学业上为我创造良好条件，使得我能够在外地实习的同时顺利完成学业。

最后，感谢我的父亲母亲，是他们几十年如一日的默默关怀、鼓励和支持，使得我一直坚强勇敢地朝着我的理想前进。

再次真诚感谢所有关心支持我的人，谢谢！

参考文献

1. Christina Warrender, Stephanie Forrest, Barak Pearlmutter. Detecting intrusions using system calls: Alternative data models[A]. In: Proc of IEEE Symp on Security and Privacy, Oakland,California. 1999, 133-14

2. Christopher M. King, Curtis E. Dalton & T. Ertem Osmanoglu. Security Architecture[M]. McGraw-Hill, 2001

3. Eric Maiwald, Network Secrity[M]. McGraw-Hill, 2001

4. Forrest S, Hofmeyr S A. Immunology as information processing[A]. In: Design Principles for the Immune System and Other Distributed Autonomous Systems[C].Segel L A, Cohen I eds, Santa Fe Institute Studies in the Sciences of Complexity. New York: Oxford University Press, 2000

5. G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview[A]. In: U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining[C]. AAAI/MIT Press, Cambridge, MA, 1996.

6. Hofmeyr S A, Forrest S, Somayaji A. Intrusion detection using sequences of system calls[J]. Journal of Computer Security, 1998, 6:151-180

7. Jake Strum. Data Warehousing Technical Reference[M].机械工业出版社

8. Michael Mullins. Implementing a Network Intrusion Detection System[M], 2002

9. Mike Fiskyx, George Varghesey. Fast Content-Based Packet Handling for Intrusion Detection[C]. UCSD Technical Report CS2001-0670, 2001

10. S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaf. A sense of self for Unix processes[A]. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, Los Alamitos, CA, IEEE Computer Society Press, 1996,120-128.

11. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process of extracting useful knowledge from volumes of data[J]. Communications of the ACM, 1996, 39(11):27-34

12. U. Fayyad, D. Haussler, and P. Stolorz. Mining scientific data[J]. Communications of the ACM, 1996, 39(11):51-57

13. Wenke Lee. Developing Data Mining Techniques for Intrusion Detection: A Progress Report.Computer Science Department[J], North Carolina State University

14. Wenke Lee, Sal Stolfo. Data mining approaches for intrusion detection[C]. In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98). San Antonio, TX, 1998

15. Wenke Lee, A data mining framework for constructing features and models for intrusion detection systems, PhD Thesie. Columbia University, 2000

16. Wenke. Lee, S. J. Stolfo, andKW. Mok. Mining in a data-flow environment: Experience in network intrusion detection[A]. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery&Data Mining (KDD-99)[C], 1999

17. William Stallings. Network Security Essentials: Applications and Standards. Prentice-Hill[M], 2000

18. 白洁英.基于数据挖掘的入侵检测系统的设计与实现[D] .中国科学院沈阳计算技术研究所，2002

19. 陈建国.使用数据挖掘技术的入侵检测模型构建[D] .上海交通大学，2003

20. 陈远春.信息安全检测鉴别监控技术与系统安全性能评估分析标准实用手册[M] .北京：人民出版社，2002

21. 戴青云.数据挖掘在网络入侵检测中的应用[D] .东南大学，2002

22. 韩东海,王超，李群系编著.入侵检测系统实例剖析[M] .北京：清华大学出版社，2002

23. 何险峰.基于数据挖掘技术和智能体技术的入侵检测系统[D] .电子科技大学，2003

24. 胡道元，闵京华.网络安全[M] .清华大学出版社，2004

25. Jiawei Han.数据挖掘：概念与技术[M] .范明等译，北京：机械工业出版社，2001

26. Jiawei Han, Micheline Kamber.数据挖掘概念与技术[M] .机械工业出版社，2001

27. 刘应玲.基于数据挖掘的入侵检测系统的研究[D] .合肥工业大学，2003.6

28. 刘水.防火墙与入侵检测系统在校园网中结合应用的初探[D] .南京理工大学，2003

29. 刘卫国.基于数据挖掘的入侵检测系统研究[D] .西安交通大学，2002

30. 鹿虹丽.数据挖掘技术在入侵检测系统模型构造中的应用研究[D] .北方交通大学，2001

31. 罗皓.属性抽取在基于数据挖掘的入侵检测系统中的应用[D] .西安交通大学，2003

32. 毛宇.数据挖掘技术在入侵检测中应用方法的研究与实现[D] .东北大学，2001

33. Paul E. Proctor 著，邓琦皓等译.入侵检测实用手册[M] .中国电力出版社，2002

34. 潘俊杰.数据挖掘技术在入侵检测中的应用[D] .北京理工大学，2003

35. Rebecca Gurley Bace 著，陈明奇等译.入侵检测[M] .人民邮电出版社，2001

36. 施伟.基于数据挖掘的入侵检测研究[D] .北京理工大学，2003

37. 谭旭阳.数据挖掘技术在网络入侵检测中的应用研究[D] .西北工业大学，2003

38. 唐正军，李建华.入侵检测技术[M] .清华大学出版社，2004

39. 王清毅，张波，蔡庆生.目前数据挖掘算法的评价[J] .小型微型计算机系统，2000，75-78

40.（美）William Stallings.网络安全要素一应用与标准[M] .北京：人民邮电出版社，2000

41. 夏可，蔡碧野.数据挖掘及其发展研究.计算机工程与应用，2002.14，182-184

42. 张学旺.基于数据挖掘技术的入侵检测研究[D] .中南林学院，2003