第1章 绪 论

(小2号黑体居中)

1.1 课题的研究背景(4号黑体)

随着信息技术的飞速发展，计算机网络在全世界各个国家的政治、军事、经济以及人民的生产生活等各个领域均广泛应用，从而信息安全已经关系到人民的安全乃至国家安全。因此，信息安全产业越来越受到国家的重视。在这样一个信息化高速发展的时代，加强网络与信息安全管理是十分必要的。互联网为人民提供了方便、快捷、舒适的服务，人民在享受这种服务的同时也要考虑到信息的保密性、完整性、认证性和不可否认性，保证个人的信息安全迫在眉睫。

随着信息安全产业的日益壮大，IDS、防火墙、漏洞扫描系统、VPN等安全产品已经成为网络体系结构中的一份子，而且，大部分网络边界都是由IDS、防火墙来提供安全防护，然而这些设备会产生海量的告警日志来记录着一个又一个安全事件。显而易见，安全管理员希望能够利用这些数据提供适当的防御和响应，从而排除更多的安全隐患，来保证用户的网络安全。然而，实际上，这并不是一件很容易就做到的事情，这些海量告警日志巨大并且冗余、分散并且混乱、错漏并且单一，很难对攻击做出相应的响应，需要大量的专家知识进行分析。因此，通过对安全设备所产生的告警数据进行综合分析和处理，挖掘出各个安全事件隐藏的逻辑关系，发现攻击者的真正目的，进而对网络攻击事件进行防御和响应，实现对整个网络环境下的有效控制，大大提高安全管理员对安全事件处理的效率，对于预防安全事件所造成的重大损失具有重要的研究意义。

随着网络安全的形式逐渐严峻，单一的安全设备已经无法满足人们的需求。防火墙无法将未知的安全事件拒之墙外，IDS也存在着误报与漏报的问题，海量的告警数据更是令人头疼。因此，就需要整合更多的安全产品，组装成一个安全网络体系，来保证边界设备内部的网络安全。这样就需要对整个网络环境下的多种设备所产生的告警日志进行综合处理和分析，从而实现对整个安全事件的分析。

根据国家计算机网络应急技术处理协调中心(CNCERT/CC)2013年中国互联网网络安全报告中按攻击类型进行统计，大部分攻击都是多步攻击，比如.cn国家顶级域名服务器遭受DDOS攻击、TP-Link路由器的域名劫持攻击等等。然而分布在网络上的各个安全设备所产生的告警都是单独的攻击行为，各个设备根据单独的攻击行为可能不会发现严重的安全事件，但这整个攻击序列可能会对该网络环境构成致命的危害。如何从各个安全设备的告警日志中挖掘出多步攻击对应的攻击序列，并将该序列及其相关属性进行统一管理，是网络安全研究中重要的研究内容。

1.2 国内外研究现状

1.2.1 告警关联分析技术的概述

1993年，Agrawal等人在[1]中首先提出关联规则的概念，它反映了大量数据中项目集之间的关联或相关联系。之后，1994年， R. Agrawal等在文献[2]中给出的Apriori算法，由于Apriori算法时间和空间复杂度较大，在挖掘长频繁模式的时候性能往往比较低下，因此，在2000年，Jiawei Han等在文献[3]中提出了基于FP-Tree的FP\_growth算法。其余大多数算法都是以这两种算法为核心进行扩展的，但这些算法大部分都是对布尔类型的事务数据库进行关联分析，挖掘出用户感兴趣的知识模式，并没有对安全事件关联分析关注得太多。追溯到告警关联分析技术研究的起源，最开始是研究分布式入侵检测告警协同分析方法，到了1999年，该项研究已经成为了信息安全领域的研究热点。告警信息关联的目的[4]就是通过对相关信息进行组合，从而得到比单个数据元素或数据片段更多的信息，进而监控目标系统遭受的攻击并做出相应的响应。网络环境中配置的各种边界设备则是关联分析研究的对象，比如：防火墙、IDS等，它们从不同的位置和角度监控该网络环境下的数据信息，比如：告警日志信息、用户特征信息等。通过对告警数据的分析，对该攻击在该网络环境下的危害等级进行判断，并采取相应的解决措施进行修复。

1.2.2 多步攻击告警关联分析方法

对攻击步骤的关联研究，首先要对数据进行聚合分析，以分布式拒绝服务攻击为例，如果需要进行更深层次的关联分析，就需要把逻辑上各个相关的攻击动作进行关联，进而对整个攻击情况进行把控[5]。据了解，如今的绝大部分网络攻击都不是单一的，而是相关联的。通常情况下，前一个攻击动作都是为后一个攻击动作做准备的。目前，告警关联分析方法可以分为以下两类：

1、通过告警信息之间的已知关系来对告警信息进行分析和处理的办法

2、不需要通过告警信息之间的已知关系来对告警信息进行分析和处理的办法

虽然上面的分类方法比较粗略，但却清晰的描述出了告警关联分析技术的发展历程。第一种办法与误用检测相似，第二种方法则与异常检测相似。如今，比较常用的关联分析有如下几种：

1、通过告警之间的属性相似度进行关联

在文献[6]中，作者提出了采用一种基于告警属性间相似性的统计方法来发现告警之间的关联；在文献[7]中，作者提出了一种运用数据挖掘技术来对相似告警进行分类的方法；在文献[8]中，将检测端口扫描攻击的启发式方法应用到了告警关联分析当中。这种关联利用设定门限的方式确定告警之间相同或相关属性有多大关联，通常以概率的形式表示，计算阈值比较困难，因此，它不可能发现告警所对应攻击间的因果联系，也不可能完整全面的理清攻击事件之间的关系。因此，该方法目前已经不是告警关联研究分析的主流。

2、分析单个攻击的先决条件和后续结果进行关联

美国北卡罗莱那州立大学 Ning]和JIGSAW 技术[9等研究者提出的告警关联方法[10][11]都属于这类告警关联方法。这两种方法都是将单个攻击的先决条件和后续结果进行关联，并将一系列的攻击用序列的方式表示出来，这种方法的优点是用起来很灵活，但它的缺点就是对数据的分析过于复杂，对于海量的攻击数据分析起来就会显得力所不能及了。

3、利用告警之间的已知关系进行关联

M-Correlator 模型[12]和 M2D2 模型[13]利用了告警之间的已知关系进行告警关联的方法。M-Correlator 首先把告警信息转换成它自己统一定义的内部事件报告格式，然后使用事件类型的脆弱程度以及受保护网络的相关知识来对这些告警进行处理。M2D2 是一种用于安全信息表示和关联处理的信息模型。该模型包含了四种类型的信息：信息系统属性、脆弱程度、安全工具及事件和告警。

4、利用多步攻击告警信息的序列进行关联

通过挖掘告警信息中安全事件类型的频繁模式[14-18]，建立起攻击序列的关联规则，利用这些关联规则对安全事件进行序列关联，该方法的难点在于难以挖掘出精准的攻击序列。通过对数据挖掘领域的经典算法(Apriori、AprioriAll和AprioriSome[19-20])进行改造，使其更适合的应用于告警信息关联分析领域，挖掘出更为精准的攻击序列，是研究告警日志关联分析的一种简单有效的解决方法。

1.2.3 多步攻击分析算法分类

通过上述多步攻击告警关联分析方法，可以总结出以下几种算法：

1、概率方法

2、聚类方法

3、学习方法

4、推理方法

5、其它方法

一般均将上述算法混合搭配进行优化，同时也包括了应用于海量数据的挖掘算法[21]。

1.2.4 多步告警关联分析的其它研究

除了上述的研究之外，告警日志关联分析领域还有许多其它的研究。比如：Kang等人提出了一个网络安全事件关联分析方法框架[22]，Goldman等人提出了一个SCYLLARUS的IDS告警关联分析系统[23]，Yu等人提出了一个叫做TRINETR的IDS告警关联分析系统[24]。周杰提出了网络攻击预警系统的研究[25]。张连华提出了基于层次的智能告警关联分析模型研究[26]。王豪提出了多源监控告警数据相关性分析的研究与应用[27]。

1.3 课题研究意义

随着信息安全产业的飞速发展，攻击手段越来越成熟，网络安全设备与日俱增，每天都会产生海量的告警信息，这些告警信息完全由人工来处理的话，将会付出巨大的代价而且困难重重。如今，对于一种入侵行为的识别以及对整个网络环境下的安全态势进行掌控迫在眉睫，保障整个网络环境下的安全刻不容缓。因此，急需一套系统来辅助安全管理员对告警数据进行整理与分析。目前，大部分攻击都不是孤立的，大多数是一套攻击序列，如何将这些孤立的攻击关联起来挖掘出攻击序列，并采取相应的措施分析攻击者的意图，是本文研究的主要内容。

对告警数据进行关联分析，可以解决以下问题：

1、误报率减少。大部分错误告警信息都是单独出现的，通过对安全事件进行关联，得到攻击序列，可以将这些错误的告警信息过滤出去。

2、漏报率减少。对于所有设备产生的告警日志，每一条安全事件都进行关联，从而准确的发现新型攻击。

3、正确的识别攻击目的。如今，大部分攻击都是多源的多步攻击，复杂的攻击组合可能产生一种新型的攻击，根据挖掘出的攻击序列，可以很好的反映出攻击者的攻击目的。从而提高网络安全水平。

本文主要研究多源设备产生的告警日志的关联分析，挖掘出隐藏的攻击序列，发现攻击者的攻击意图，掌握整个网络安全态势信息。在数据关联方面，如何提高关联数据的准确性、直观性，还需要有更多的研究。因此，本文研究的内容具有重要的研究意义。

1.4 论文的主要研究内容和组织结构

本文通过对边缘设备告警日志的分析，结合国内外对关联分析的研究状况，试图对多源告警日志关联分析进行深层次的研究，设计了基于动态更新时间阈值以及量化分析的关联分析算法，并实现了多源告警日志关联分析系统。主要研究内容如下：

1、对现有的关联分析技术进行研究和总结；

2、对IDS和防火墙的告警信息进行格式统一处理；

3、针对某单位的告警日志信息，设计了一套多源告警日志关联分析系统框架；

4、对动态更新时间间隔阈值的算法进行研究；

5、设计了基于Apriori系列的序列模式挖掘算法；

6、对安全序列属性进行量化分析，过滤掉小于最小概率的数据，从而得到更精准的攻击序列信息；

本文将分为六章进行介绍，各章的组织结构如下：

第一章是绪论，主要介绍了本课题的研究背景以及意义，叙述了告警关联分析在国内外的研究方法和现状。

第二章为相关技术，主要介绍开发多源告警信息关联分析系统所用到的各种技术，包括Python，Django，Mysql数据库，REST架构，Nginx服务器。

第三章主要研究基于Apriori系列的关联分析算法，在此基础上研究并设计基于序列模式的告警关联分析算法，动态更新时间间隔阈值算法，基于告警信息相关属性的量化分析算法，最终得出最后的攻击序列信息。

第四章主要介绍针对于某单位的多源告警信息的关联分析系统的设计与实现。

第五章主要是应用国际上标准的darpa2000数据集对本系统设计的关联分析算法进行测试与评价，收集某单位产生的IDS和防火墙的告警日志信息对本系统进行关联分析与评价。

第六章为总结与展望。主要介绍本文的工作总结以及对未来的展望。

第2章 相关技术

2.1 Python技术

2.1.1 Python概述

Python，是一种面向对象、解释型计算机程序设计语言。自从1991年诞生以来，Python现在已经成为最受欢迎的动态编程语言之一。由于拥有大量的Web框架，最近几年非常流行使用Python进行网站建设工作。

Python具有清晰的语法结构。默认安装的Python开发环境已经附带了很多高级数据类型，如：列表、元组、字典、集合、队列等。使用这些数据类型使得实现抽象的数学概念非常简单。由于Python语言的简洁、易读以及可扩展性，有着丰富的标准库，其标准库可以帮助开发者处理各种工作，如：图形用户界面、文件处理、多媒体、正则表达式、文档生成、单元测试、线程、数据库、网络通讯、网页浏览器、CGI、FTP、电子邮件、XML、HTML、WAV文件、密码系统、Tk和其它与系统有关的操作。Python的在设计上坚持了清晰划一的风格，这使得Python成为一门易读、易维护、用途广泛的语言。Python的语法独特，它使用缩进来区分语句的关系，而不再使用一对花括号{}，这样可以强制程序员养成良好的编程习惯。由于Python是解释性语言，在执行时，会首先把源代码编译成Python的byte code（字节码），然后再由Python Virtual Machine（Python虚拟机）来执行这些编译好的字节码，下次再执行时，无须再编译源代码而是直接执行编译好的字节码文件。并且Python还可以以交互模式运行，比如主流操作系统Unix、Linux、Mac、Windows都可以在命令模式下直接运行Python交互环境，直接下达操作指令即可实现交互操作。

2.1.2 Python在科学计算领域中的特色

在科学和金融领域，Python语言得到了广泛应用。SciPy和NumPy等许多科学函数库都实现了向量和矩阵操作，这些函数库增加了代码的可读性，学过线性代数的人都可以看懂代码的实际功能。另外，科学函数库SciPy和NumPy使用底层语言（C和Fortran）编写，提高了相关应用程序的计算性能。近年来，在数据分析和交互、探索性计算以及数据可视化等方面，Python有不断改良的库（主要是pandas），使其成为数据处理任务的一大替代方案。

Java和C等强类型程序设计语言也有矩阵数学库，然而对于这些程序设计语言来说，最大的问题是即使完成简单的操作也要编写大量的代码。程序员首先需要定义变量的类型，对于Java来说，每次封装属性时还需要实现getter和setter方法。另外还要记着实现子类，即使并不想使用子类，也必须实现子类方法。为了完成一个简单的工作，必须花费大量的时间编写很多无用冗长的代码。Python语言则与Java和C完全不同，它清晰简练，而且易于理解，即使不是编程人员也能够理解程序的含义，而Java和C对于非编程人员则像天书一样难于理解。

Python语言是高级编程语言，程序员可以花费更多的时间处理数据的内在含义，但无需花费太多精力解决计算机如何得到数据结果。Python语言使得人们很容易表达自己的目的。

2.2 Django框架

Django是一个基于MVC构造的框架。但是在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。它们各自的职责如表2-1所示：

**表2-1 MVT模型**

|  |  |
| --- | --- |
| 层次 | 职责 |
| 模型（Model），即数据存取层 | 处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。 |
| 模板(Template)，即表现层 | 处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。 |
| 视图（View），即业务逻辑层 | 存取模型及调取恰当模板的相关逻辑。模型与模板之间的桥梁。 |

2.3 Mysql数据库

2.3.1 Mysql的概述

Mysql数据库是一款非常优秀的关系型数据库，由C/C++编写，现在已经是世界上最流行的数据库之一。全球最大的网络搜索引擎公司Google使用的就是Mysql数据库。国内很多大型的网络公司也选择Mysql数据库，如网易、新浪等。这都证明了Mysql数据库强大的生命力。Mysql不仅可以在Windows系列的操作系统上运行，还可以在UNIX、Linux和Mac OS等操作系统上运行。因为很多网站都选择UNIX、Linux作为网站的服务器，所以Mysql的跨平台性保证其在Web应用方面的优势。虽然微软公司的SQL Server数据库是一款很优秀的商业数据库，但是其只能在Windows系列的操作系统上运行。因此，Mysql数据库的跨平台性也是一个很大的优势。

2.3.2 Mysql的特性

1、开放性：它所使用的语言SQL以ANSI SQL2为基础，这个数据库引擎可以运行在多个平台上，包括Windows、Mac OSx、Linux、FreeBSD和Solaris等。

2、应用支持：Mysql为几乎任何编程语言提供了一个API。

3、跨数据库连接：可以连接不同数据库的表以建立Mysql查询。

4、外连接支持：Mysql支持ANSI及odbc的左外连接和右外连接。

5、Mysql廉价而且快速。

* 1. REST架构

2.4.1 REST架构简介

REST（Representational State Transfer）是一种轻量级的Web Service架构风格，其实现和操作明显比SOAP和XML-RPC更为简洁，可以完全通过Http协议实现，还可以利用缓存Cache来提高响应速度，性能、效率和易用性上都优于SOAP协议。REST架构规范如下：

1、统一接口

REST 架构风格的核心特征就是强调组件之间有一个统一的接口，这表现在REST世界里，网络上所有的事物都被抽象为资源，而REST就是通过通用的链接器接口对资源进行操作。这样设计的好处是保证系统提供的服务都是解耦的，极大的简化了系统，从而改善了系统的交互性和可重用性。并且REST针对Web的常见情况做了优化，使得REST接口被设计为可以高效的转移大粒度的超媒体数据。

2、分层系统

分层系统规则的加入提高了各种层次之间的独立性，为整个系统的复杂性设置了边界，通过封装遗留的服务，使新的服务器免受遗留客户端的影响，这也就提高了系统的可伸缩性。

3、无状态

虽然REST包含无状态性（statelessness）的观念，但这并不是说暴露功能的应用不能有状态。事实上，在大部分情况下这会导致整个做法没有任何用处。REST要求状态要么被放入资源状态中，要么保存在客户端上。或者换句话说，服务器端不能保持除了单次请求之外的，任何与其通信的客户端的通信状态。这样做的最直接的理由就是可伸缩性。此外还有很重要的一点，无状态约束使服务器的变化对客户端是不可见的，因为在两次请求中，客户端并不依赖于同一台服务器，这样做的好处是，即使服务端发生了一些变化，对客户端而言是透明的。由于没有了上下文的约束，做分布式和集群的时候就更为简单。

4、标准方法

REST是基于Http协议的，任何对资源的操作行为都是通过Http协议来实现。以往的Web开发大多数用的都是Http协议中的GET和POST方法，对其它方法很少使用，这实际上是因为对Http协议认识片面的理解造成的。Http不仅仅是一个简单的运载数据的协议，还是一个具有丰富内涵的网络软件的协议。Http把对一个资源的操作限制在4个方法以内：GET, POST,PUT和DELETE，这正是对资源CRUD操作的实现。由于资源和URI是一一对应的，执行这些操作的时候URI是没有变化的，这和以往的 Web开发有很大的区别。正由于这一点，极大的简化了Web开发，也使得URI可以被设计成更为直观的反映资源的结构，这种URI的设计被称作 RESTful的URI。这为开发人员引入了一种新的思维方式：通过URL来设计系统结构。当然了，这种设计方式对一些特定情况也是不适用的，也就是说不是所有的URI都可以RESTful的。

2.4.2 Django REST framework 简介

Django REST framework 承袭django的结构，直接提供Browsable API,易用性很高。最重要的一点是该框架对于resource,serializer,renderer/parser,view和response的定义很清晰，有很符合Django的MVT模式（比如：它的view就是包装了Django的View实现的）。该框架对于认证和授权也有很好的支持，并且内置了一系列的Mixin，可以随意组装。因此，在Django使用该框架可以说是顺乎自然。

2.5 Nginx服务器

Nginx是一款高性能的HTTP和反向代理服务器。Nginx能够选择高效的epoll、kqueue、eventport作为网络I/O模型，在高连接并发的情况下，Nginx是Apache服务器不错的替代品，它能够支持高达50000个并发连接数的响应，而内存、CPU等系统资源消耗却非常低，运行非常稳定。

Nginx已经在俄罗斯最大的门户网站（Rambler Media）上运行了超过7年的时间，同时俄罗斯超过20%的虚拟主机平台采用Nginx作为反向代理服务器。在国内，已经有新浪、网易、56.com、豆瓣、迅雷在线等多家网站使用Nginx作为Web服务器或反向代理服务器。

选择Nginx服务器有如下一些理由：

1、它可以高并发连接

2、内存消耗少

3、成本低廉

4、配置文件非常简单

5、支持Rewrite重写规则

6、内置的健康检查功能

7、节省带宽

8、稳定性高

9、支持热部署

表2-2是Nginx与Apache、Lighttpd的综合对比情况。

**表2-2 Nginx与Apache、Lighttpd的综合对比**

|  |  |  |  |
| --- | --- | --- | --- |
| Web服务器 | Nginx | Apache | Lighttpd |
| 反向代理 | 非常好 | 好 | 一般 |
| Rewrite规则 | 非常好 | 好 | 一般 |
| FastCGI | 好 | 差 | 非常好 |
| 热部署 | 支持 | 不支持 | 不支持 |
| 系统压力比较 | 很小 | 小 | 很大 |
| 稳定性 | 非常好 | 好 | 一般 |
| 安全性 | 一般 | 好 | 一般 |
| 技术资料 | 很少 | 非常多 | 一般 |
| 静态文件处理 | 非常好 | 一般 | 好 |
| 虚拟主机 | 支持 | 支持 | 支持 |
| 内存消耗 | 非常小 | 很大 | 非常小 |

通过表2-2可以看出，Nginx在反向代理、Rewrite规则、稳定性、静态文件处理、内存消耗等方面，表现出了很强的优势，选用Nginx取代传统的Apache服务器，将会获得多方面的性能提升。

2.6 本章小结

本章主要介绍了开发多源告警日志关联分析系统中会使用到的技术、框架以及系统使用的体系架构。

第3章 基于攻击行为序列模式挖掘与量化分析算法实现

目前，网络采集的告警信息都是具有特定格式的短消息，用于提示网络或系统故障。一条告警包括消息创建和接收时间，故障描述，告警严重性，设备状态和配置，多个告警的聚合信息等。由于告警仅仅描述了故障的发生，并不说明故障发生的原因；网络管理员每天收到大量告警信息，关键信息淹没其中；网络的异构和多样性加大了网络管理员融合多源告警的难度。因此，本文基于Apriori系列算法的思想，设计了基于攻击行为序列模式的挖掘算法。针对本问题，算法采用动态时间间隔阈值做安全事件序列划分，设计量化分析算法对关联结果进行评估，挖掘出更加精准的攻击序列信息。

3.1 基于序列模式的告警关联算法研究

序列模式挖掘主要是根据某个或某几个事件序列在一个时间段内出现的频程度来判断是否构成一个序列模式，如果出现得非常频繁，则可以将这些按照时间发生先后顺序构成一个规律，这样的规律便是序列模式。可以用来对未来事件做出一些辅助的预测。

序列模式与关联规则的发现方法大致相同，序列模式的发现可视为在时态数据库中发现关联。不过，关联规则仅仅是发现事务内部的模式，而序列模式则是发现事务之间的模式。由于安全事件中隐含了构成多步攻击场景的安全事件发生序列模式，因此需要研究攻击场景的时间窗口，利用攻击场景时间窗口来把安全事件数据库转换成候选攻击序列集合。目前主要的序列模式挖掘算法包括：AprioriAll、SPADE、PrefixSpan、SPAM等算法。

3.2 Apriori类序列模式挖掘算法

Agrawal 和 Srikant 对序列模式挖掘给出的定义如下：给定一个序列数据库，其中每个序列由元素集合组成，每个元素为一个项目集，同时还给定一个最小支持度，序列挖掘就是找出所有的频繁子序列，即在序列库中的发生次数不低于最小支持度的子序列。

给定支持度阈值x， 如果序列a在序列数据库中的支持数不低于x， 则称序列a为序列模式，长度为l的序列模式记为l-模式。在一个序列集中如果序列s不包含于任何其他序列中，则称序列s为最大或极大的(maximal)，又将最大的序列模式称为序列模式。

Agrawal 等人提出的挖掘方法分为五个步骤：

1、挑选数据：从数据库中挑选相关的记录和属性，并转换成时间序列；

2、构造频繁项目集：在时间序列中寻找频繁项目集，自动生成频繁l-序列；

3、转换：将时间序列中的每一事务转换为事务中包含的频繁项目集，并用整型数值来表示；

4、发现序列模式：依据频繁l-序列，采用类 Apriori 算法，迭代构造所有的频繁序列；

5、寻找最大序列模式：从频繁序列集合中挑选最大序列模式。

根据步骤4和步骤5的不同， Agrawal 提出了一组算法： AprioriAll、 AprioriSome 和 DynamicSome。其中 AprioriAll 逐层迭代计算所有的频繁序列，最后从中剔除非最大（maximal）序列模式。算法只对最大序列模式感兴趣，因此，既要尽量避免记录那些包含在更长的频繁序列中的子序列，又要减少不具有最小支持度的较长序列的生成。AprioriSome 和DynamicSome 都是基于这种考虑的算法，两者的区别是产生候选序列的方法不同，前者仅依据上次生成的频繁序列构造候选频繁序列集，后者使用以往已经生成的频繁序列以及从数据库中读取的序列产生候选频繁序列集。

AprioriAll算法源于Apriori算法，把Apriori算法的基本思想扩展的序列挖掘中。即对长度为 k 的候选序列 Ck 中的每一个序列计算支持度，如果符合最小支持度则产生长度为k 的大序列Lk。然后再利用AprioriAll-generate产生下一个Ck+1，再找出Lk+1，…，如此循环，直到找到所有的大序列为止。

AprioriSome 算法可以看是AprioriAll 算法的改进。算法分为两个阶段：Forward和 Backward。Forward 阶段只对一定长度的序列计数，由 next(k)决定下次需要计数的序列长度。Backward阶段对那些在Forward阶段跳过了的序列计数。

3.3 多步攻击的告警挖掘方法

序列模式挖掘算法是从多个序列中挖掘出频繁发生的最大子序列。传统序列模式算法针对的是交易数据库，每个顾客购买商品的行为形成一个对应的序列。而安全事件数据库不同于交易数据库，安全事件不是以序列集的形式存放在数据库中。但是安全事件中隐含了构成多步攻击场景的安全事件发生序列模式，在攻击场景中具有时间窗口的概念，因此，可以利用攻击场景时间窗口来把安全事件数据库转换成候选攻击序列集合。多步攻击的告警算法的目标是从候选攻击序列集合中挖掘出多步攻击行为发生模式。

原始安全事件告警经过聚合和确认处理后，告警数量大大减少，告警质量得到明显提高。而攻击行为模式挖掘算法就是针对这些高级安全事件告警提出的。从高级安全事件告警数据库中挖掘攻击行为模式的思想是基于多步攻击的攻击步骤的发生具有一定的顺序和发生模式的特点而提出的。通过对多步攻击实例的研究和观察得知，某一种特定的多步攻击的攻击步骤具有一定的发生模式，并且这些攻击步骤在某个时间间隔内完成，往往上一步攻击行为的成功才能导致下一步攻击行为的发生。例如在一个DDoS攻击中，攻击者必须先在有漏洞的主机上安装DDoS守护进程，才能再指导这些守护程序发动攻击。也即，攻击者必须到达某一个状态，才能执行下一步攻击行为。攻击者的攻击步骤的发生是具备一定顺序和模式的。 在挖掘多步攻击行为发生模式时，关心的是攻击行为发生序列，所以主要考虑告警中的攻击类型属性。

3.4 基于攻击行为序列模式的挖掘算法实现

告警日志记录了大量的攻击信息，研究序列模式的挖掘算法是研究告警日志关联分析的基础。如今攻击类型与日俱增，攻击形式各式各样，仅凭专家知识来判断攻击序列的难度越来越大，然而，频繁的告警序列确是一个研究攻击序列的很好的模板来源。本文基于Apriori算法以及AprioriAll算法的设计思想，结合现有多源攻击的模式，研究了针对于频繁告警序列的序列模式挖掘问题，设计出了针对于该问题的挖掘方法，并针对分析出的安全序列的相关属性进行量化分析，从而得出更加准确的与攻击序列相关的信息。

关联规则是一种在大规模数据下寻找有趣关系的任务。这些关系可以有两种形式：频繁项集或者关联规则。频繁项集(frequent item sets)是经常出现在一块的物品的集合，关联规则(association rules)暗示着两种或多种物品之间可能存在着很强的关系。然而，本文中挖掘算法的目的是在大规模告警日志中挖掘出符合一定专家知识的攻击序列。相关的基本概念日如下：

定义3.1、原始告警序列是按时间排序的告警集合，一个告警集合Es可表示为<e1,e2,e3, …,en>,ti(i=1,2,3, …,n-1)为原始告警Es的时间， (i=1，2，…,n-1)相邻告警之间的时间间隔。

定义 3.2、动态时间窗口TW，令τi为相邻告警之间的时间间隔， 为时间间隔平均值， 为系数，TW可以表示为：

定义3.3、序列的划分是指根据动态时间窗口将原始告警序列分割成若干个在一定时间间隔内的小序列s，一个小序列s可以用<s1,s2,s3,…,sn>来表示，这里的s可以作为告警集合Es的一个项集。

定义3.4、一个项集的支持度（support）被定义为数据集中包含该项集的记录所占的比例。比如某个环境下的告警日志，如表3-1所示：

表3-1 某个环境下的告警日志

|  |  |
| --- | --- |
| 时间段序号 | 攻击序列 |
| 0 | FW-NAT、dos攻击 |
| 1 | dos攻击、SNMP\_缺省口令[public]、SNMP\_登录尝试、FW-NAT |
| 2 | FW-NAT、dos攻击、FW-NAT |
| 3 | FW-NAT、dos攻击、ARP\_IP地址欺骗 |
| 4 | FW-NAT、流量信息、FW-NAT、dos攻击 |

从表一中可以得到，｛dos攻击｝的支持度为1。而在这5条交易记录中有4条的序列为｛FW-NAT,dos攻击｝，因此｛FW-NAT,dos攻击｝的支持度为4/5。支持度是针对项集来说的，因此可以定义一个最小支持度，而只保留满足最小支持度的项集。

定义3.5、序列的长度指的是该序列包含的项集的个数。例如一个有k长度的序列，我们就把该序列称为k-序列。

定义3.6、频繁项集指的是项集s的相对支持度满足预定义的最小支持度阈值，那么我们就把s成为频繁项集。如果某个项集是频繁的，那么它的所有子集也是频繁的。比如说｛0，1｝是频繁的，那么｛0｝、｛1｝也一定是频繁的。反之，如果一个项集是非频繁的，那么它的所有超集也是非频繁的。

3.4.1 算法设计

本文是针对频繁告警序列的序列模式挖掘问题所设计的算法AprioriAI，算法的主要步骤如下：

通过动态的时间窗口，生成若干个在某段动态时间内的攻击序列的集合。

输入：原始告警序列Es

输出：若干个在某段时间内的攻击序列s1，s2,s3,…,sn

for i in range(len(data))

∆t=t\_(i+1 )- t\_i i=0,1,2,3,…,length(data)-1 //计算相邻告警日志之间时间戳之差

if t >= ∆t

events.append(data[i+1][1])//将攻击类型添加至列表当中

timeArray.append(∆t)//将两者时间戳之差加入时间戳列表

t = timeSet(sumTime,n,timeArray)//调用动态生成时间阈值函数，动态生成时间阈值方法如定义3.2所示，并将值附给t

else：

result.append(events)//将所得的攻击序列列表si存入到一个大的整体列表Es当中，同时所需变量进行初始化。

上述算法的关键是序列数据集S的产生，下面将进行频繁项集挖掘阶段，这里相当于用了一次apriori算法，找出大于给定支持度的频繁项集

输入：序列数据集S

输出：大于给定支持度的频繁项集

for tid in D:

for can in Ck:

iscan,can = isSubset(can,tid)//对数据集中每一个候选者进行序列比较

if iscan == True://如果成功匹配，则进行计数

if not ssCnt.has\_key(can):

ssCnt[can] = 1

else:

ssCnt[can] += 1 //统计序列数据集中每一个序列告警s的个数

for key in ssCnt:

support = ssCnt[key]/numItems //计算每一个候选者的支持度

if support >= minSupport: //如果有满足最小支持度的候选者，则添加到列表当中，进行保存

retList.insert(0,json.loads(key))

supportData[key] = support//记录每一个候选者的支持度，便于统计

//retList为大于给定支持度的频繁项集

上述算法的关键是产生了大于给定支持度的频繁项集，下面要对这些频繁项集进行最大序列化阶段

输入：上述得到的retList

输出：具有最大序列化的频繁项集

while (len(L[k-2]) > 0):

Ck = aprioriGen(L[k-2],k) //产生候选集Ck，对两个相同的序列串进行合并，不同的序列串进行添加

Ck = json.dumps(Ck)

Lk,supK = scanD(D,Ck,minSupport)//采取同样的方式计数，大于最小支持度的保留下来

supportData.update(supK)

L.append(Lk)

k += 1 //k值依次递增

上述算法的关键是得到了最大序列的频繁项集，也就是在一定时间内最长的攻击序列，下面进行分析上述的aprioriGen()函数，即对两个序列串进行处理的方法

输入：各个阶段的频繁项集以及阶段数

输出：合并后的序列

for i in range(lenLk):

for j in range(0,lenLk):

L1 = list(Lk[i])[:k-2]

L2 = list(Lk[j])[:k-2]

L1.sort()

L2.sort()

list1 = list(Lk[i])

list2 = list(Lk[j])//对两个序列进行格式统一处理

if L1 == L2://如果相同的话，则删除一个序列

if list1 == list2:

del list1

else:

list1.append(list2[-1])//如果不同的话，则序列1添加一个与序列2不同的元素

retList.append(list1)//将序列1添加到列表中

上述算法就得出了合并后的序列

3.5 基于攻击序列属性的量化分析评估算法实现

所谓量化分析就是将一些不具体的，模糊的因素用具体的数据来表示，从而达到分析比较的目的。每条攻击序列信息都包括以下属性：攻击时间、攻击源地址、攻击目的地址、攻击源端口、攻击目的端口。可以利用这些信息排除调不属于同一攻击场景的告警事件，从而过滤掉攻击序列，进而使得剩下的攻击序列更加的准确。

3.5.1 攻击序列属性的量化分析定义

根据两条攻击序列的属性信息可以判断出同一个攻击场景的攻击前后步骤的关联性是否可靠。

定义3.7、攻击序列属性的关联度Cor(hi,hj)。Cor(hi,hj)表示相邻的两条告警属性hi,hj之间的关联程度。令hi,hj分别为相邻的两条告警信息，分别由n个属性x1,x2,x3,……xn和y1,y2,y3,……yn来表示，那么它们的关联程度Cor(hi,hj)可以定义为它们属性关联度的函数： (3.5-1)

其中wij为权值，是由个人的经验来决定的，要根据实际情况进行调整。参与属性关联度分析的目前包括：攻击时间、源攻击地址、目的攻击地址、源攻击端口、目的攻击端口 共五个属性。因此，需要5\*5的权值矩阵。如表3-2所示：

**表3-2 攻击序列属性权值矩阵**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| i(前一步)\j(后一步) | SrcIP | DstIP | SrcPort | DstPort | OccurTime |
| SrcIP | 0.2 | 0 | 0 | 0 | 0 |
| DstIP | 0.2 | 0.2 | 0 | 0.1 | 0 |
| SrcPort | 0 | 0 | 0.05 | 0 | 0 |
| DstPort | 0 | 0 | 0 | 0.05 | 0 |
| OccurTime | 0 | 0 | 0 | 0 | 0.2 |

表3-2给出了5\*5矩阵的各个属性的权值，以下对各个属性的权值进行说明：

1、前一步的SrcIP跟后一步的SrcIP可能存在着关联，可以判断攻击序列是从一台主机中发出还是一个网段中发出或者没有关联。

2、前一步的DstIP跟后一步的SrcIP可能存在着关联，前一步的DstIP跟后一步的DstPort可能存在着关联，这些关联可能表现出攻击序列中某两条攻击是否存在着因果关系。

3、前一步的SrcPort跟后一步的SrcPort，前一步的DstPort跟后一步的DstPort，之间存在的关系较弱，所以它们之间的关联度的权值较低。

4、一个攻击的发生时间可以反映出攻击序列中某两条告警存在的关系，可以根据时间来判断某两条攻击是否存在这关联。

3.5.2 攻击序列属性的量化分析算法设计

表3-3对攻击序列属性间的关联函数进行说明

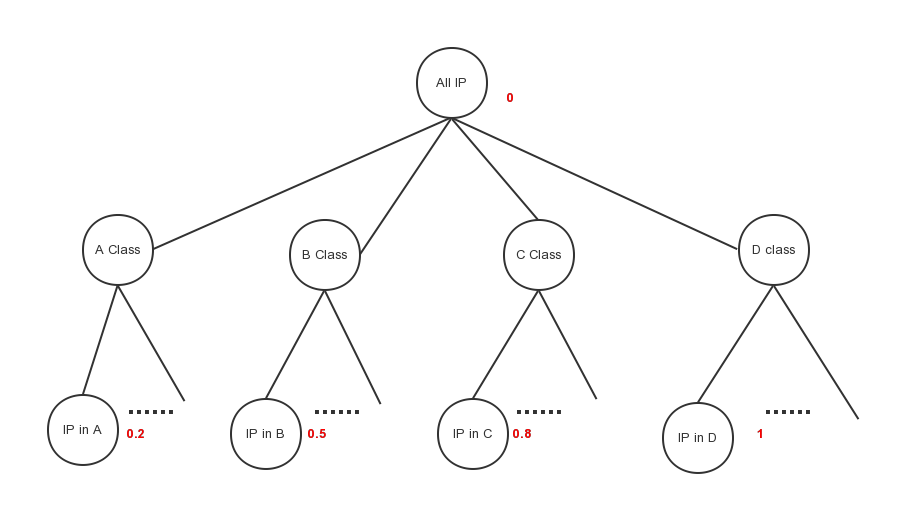
**表3-3 攻击序列属性关联说明**

|  |  |
| --- | --- |
| 关联函数 | 函数说明 |
| Cor(SrcIPi,SrcIPj) | 前一个SrcIP跟后一个SrcIP之间的关系 |
| Cor(DstIPi,DstIPj) | 前一个DstIP跟后一个DstIP之间的关系 |
| Cor(SrcPorti,SrcPortj) | 前一个SrcPort跟后一个SrcPort之间的关系 |
| Cor(DstPorti,DstPortj) | 前一个DstPort跟后一个DstPort之间的关系 |
| Cor(Timei,Timej) | 前一个攻击时间跟后一个攻击时间之间的关系 |

以下对量化分析的算法设计进行说明：

1、IP地址间的关联

可以对IP地址进行分类，从而来计算IP地址之间的关联关系。如图5-1所示对IP地址进行分类。



**图3-1 IP地址关系分类**

如果IP地址S1与IP地址S2同属于A类，那么计算Cor(S1,S2),根据上图所示，关联度值为0.2。

2、端口间的关联

如果前一步的源端口SrcPort1与后一步的源端口SrcPort2相同，则源端口之间的关联度为0.05，即Cor(SrcPort1,SrcPort2)为0.05，不同即为0。如果前一步的目的端口DstPort1与后一步的目的端口DstPort2相同，那么目的端口之间的关联度为0.05，即Cor(DstPort1,DstPort2)为0.05，不同即为0。

3、时间的关联

Cor(Timei,Timej)为前一步攻击发生的时间和后一步攻击发生的时间的关系的关联程度。时间属性关联度计算的定义如下：

(3.5-2)

表示两条相邻告警之间的时间之差的误差范围，单位为秒。这样得到的Cor(Timei,Timej)的取值范围也在0-1之间。

4、量化分析

通过先验知识设定一个两条告警之间的最小关联度为p，将小于最小关联度的攻击序列过滤掉。设定量化分析结果为f，wij为攻击前一步和后一步的权值，相关计算定义如下：

f为分析出来的结果，将其与p进行比较，将符合要求的攻击序列记录下来，从而得到更加精准的攻击序列。

3.6 小结

本章介绍了Apriori系列的数据挖掘算法，并根据多步攻击的特征，提出了一种应用于挖掘攻击序列的序列模式挖掘算法，运用了动态生成时间间隔阈值的思想，对现有的Apriori系列算法进行改造，并对得出的安全序列中事件的相关属性进行量化分析，从而过滤掉不准确的安全序列，使最终得出的安全序列更加的精准。

算法只针对攻击类型进行序列模式挖掘，主要研究多步攻击中的攻击序列，易于实现。实验表明，该方法能够有效的将海量告警日志中的多步攻击序列挖掘出来，再将挖掘出来的数据进行过滤提取，使得结果的准确性有较大的提升，并可应用于实时监测，具有一定的创新性。

第4章 多源告警信息的关联分析系统实现

4.1 系统简介

多源告警信息的关联分析系统主要是针对某单位的IDS、防火墙等设备产生的告警日志并对其进行高效的关联分析而设计的。主要目的是挖掘出新型的多步攻击序列，并采取有效措施进行防御，从而提高整个网络环境下的信息安全。

4.1.1 功能描述

本系统主要是针对某单位的告警数据进行关联分析，其主要功能如下：

1、告警数据预处理

对于大量的告警数据属性，需要提取一些应用于算法中关键的信息，并且利用动态生成时间间隔阈值算法，对攻击类型进行划分，为应用序列模式挖掘算法做前期的准备。

2、基于Apriori系列算法对安全事件进行关联

利用设计的序列模式挖掘算法对安全事件进行关联分析，挖掘出频繁的攻击序列，为安全管理员提供有效的参考数据

3、告警信息统计分析

通过关联分析算法，对告警日志中的安全类型进行统计分析，并以图形的形式展示出来。

4、量化分析

对告警信息进行量化分析，通过评估从而得到更加精准的攻击序列信息，并将结果显示出来，使得安全管理员可以更直观的把控整个网络环境的安全。

4.1.2 运行及开发环境

1、运行环境：ubuntu12.04 Nginx服务器

2、开发语言：Python、css、html、javascript

3、开发工具：Sublime text2

4、数据库：mysql

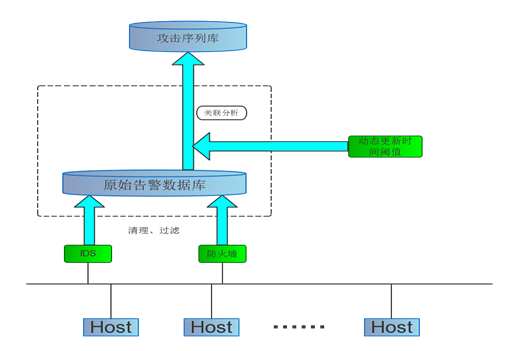
5、开发框架：Django

4.2 系统的设计与实现

系统属于Web开发，后端在Linux环境下运行，前端在浏览器中运行。本文设计的多源告警信息的关联分析系统主要包括四个模块：告警预处理模块、告警关联模块、时间间隔阈值更新模块、量化分析模块，另外还包括支持这些模块的原始告警库、攻击序列库。本文涉及的技术点包括：告警关联、告警时间间隔阈值动态更新、量化分析。下面将对系统的关键技术进行描述。

4.2.1 系统核心设计模型

如图4-1为多源告警信息的关联分析模型：



**图4-1 多源告警信息的关联分析模型**

首先从网络中采集防火墙和IDS日志告警数据，经过预处理模块将清理、过滤后的数据以统一的格式存放在原始告警数据库中，利用本来设计的关联算法将原始告警数据库中的告警数据进行关联，并将关联分析结果写入攻击序列库中。动态更新时间阈值，以适应告警日志的动态变化。

4.2.2 数据格式约定

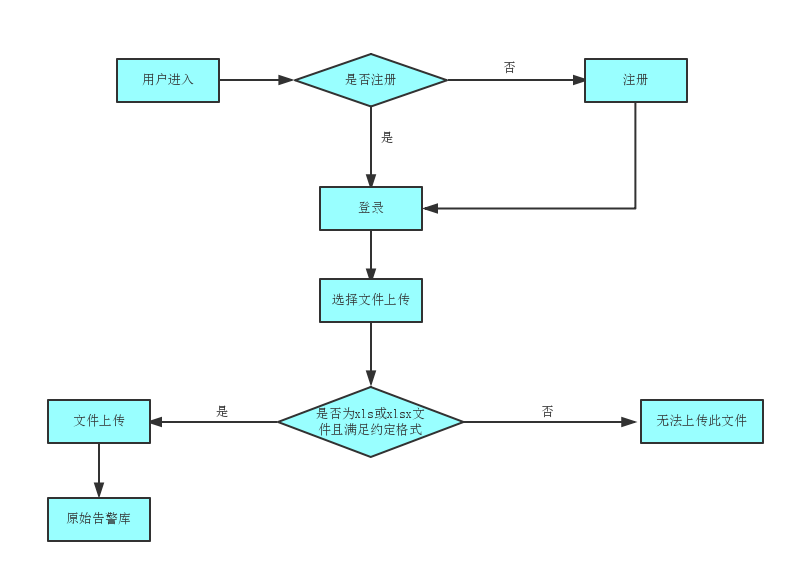
导入xls的段属性约定，格式如表4-1所示：

**表4-1 导入xls的段属性约定**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 系统类型 | 接收时间 | 等级 | 源地址 | 源端口 | 源用户名称 | 目的地址 | 目的端口 | 设备地址 | 设备类型 | 时间名称 |
|  |  |  |  |  |  |  |  |  |  |  |

4.2.3 数据导入

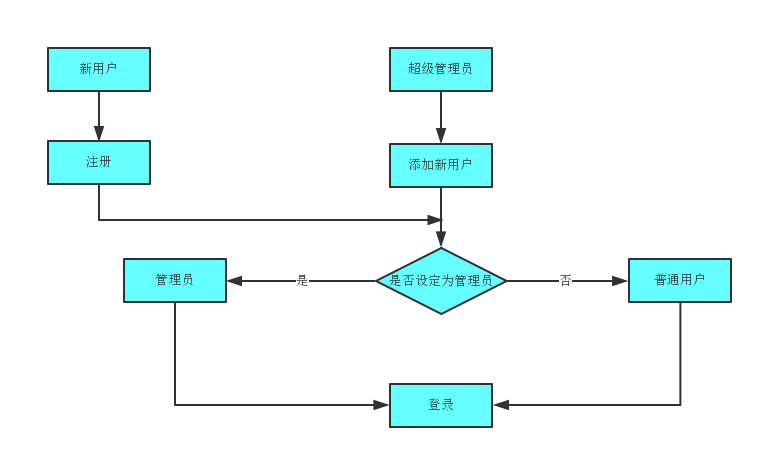
将约定好的xls文件导入数据库，具体流程如图4-2所示：



**图4-2 xls导入流程**

4.2.4 用户管理

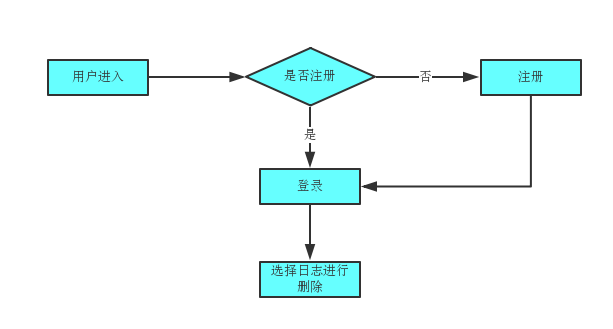
该部分主要负责用户权限的设定，用户注册后，超级管理员可以附给用户相应的权限，管理员也可以添加用户，使得用该系统的人员为安全管理员，从而保障系统内部数据的安全，具体流程如图4-3所示：

****

**图4-3 用户管理模块流程图**

4.2.5 告警日志删除

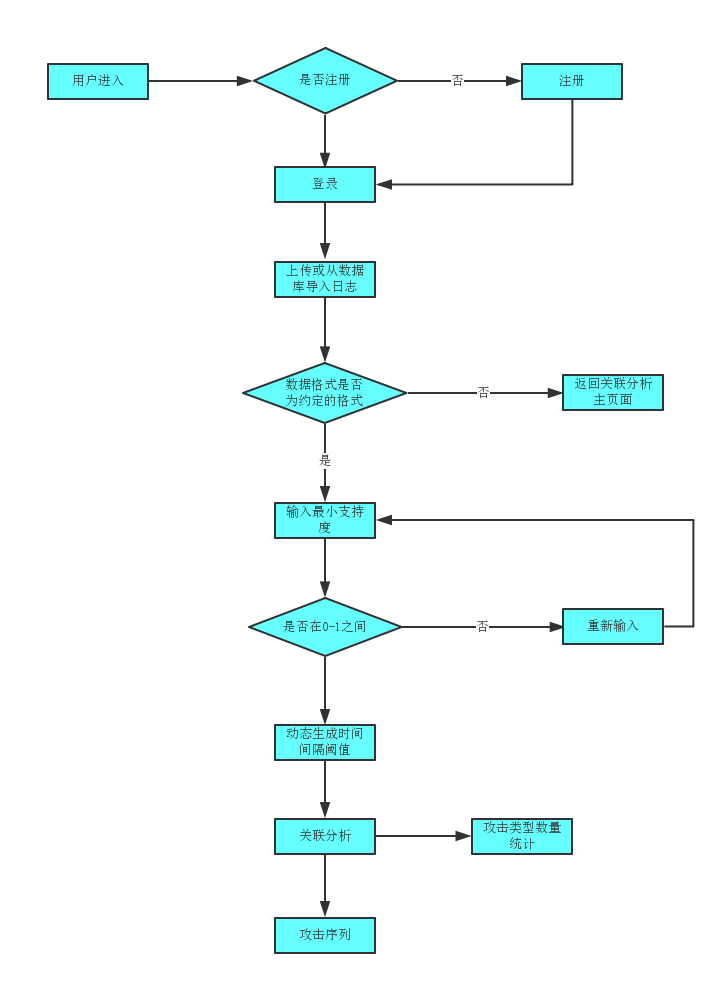
用户登录该系统时，如果不是已注册用户，首先要注册或者管理员添加用户，然后再进行登录。成功登录后，选择要删除的告警日志进行删除操作，具体操作流程如图4-4所示：



**图4-4 日志删除操作流程图**

4.2.6 关联分析

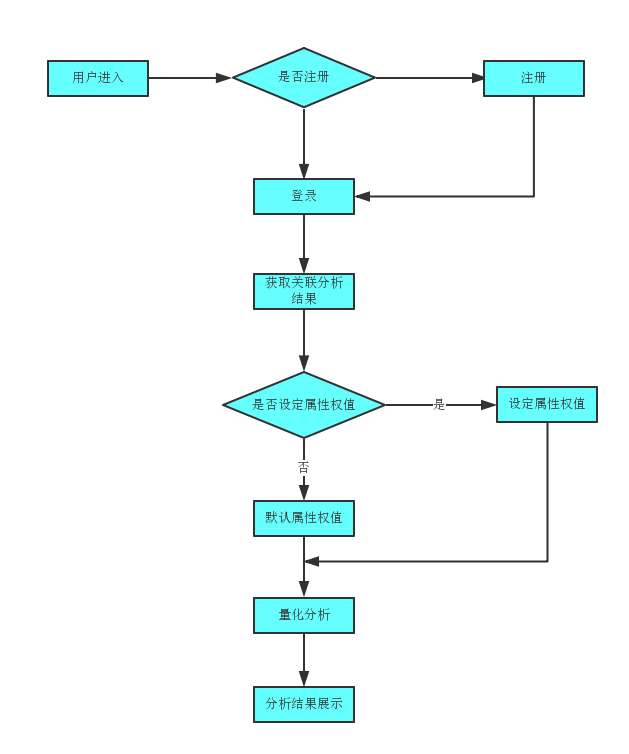
用户可以从数据库中导入数据进行关联分析，也可以重新上传日志，对该日志进行关联分析，但该日志不导入到数据库中。关联分析时，用户可以设定最小支持度，最后会给出关联结果以及攻击数量的统计，具体流程如图4-5所示：



**图4-5 关联分析流程图**

4.2.7 量化分析

将关联分析结果做进一步的量化分析，可以对关联结果的准确性做进一步的提升。用户可以设定源地址、源端口、目的地址、目的端口、攻击时间的权值，也可以使用默认的权值，权值根据影响攻击结果所占的比重分配，具体分析流程如图4-6所示：



**图4-6 量化分析流程图**

4.2.8 数据库表设计

1、系统用户表设计，如表4-2所示：

**表4-2 系统用户表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 概念名称 | 字段名称 | 字段类型 | 宽度 |
| 1 | 用户ID | id | CharField | 40 |
| 2 | 用户名 | username | CharField | 40 |
| 3 | 用户密码 | password | CharField | 40 |
| 4 | 用户邮箱 | Email | EmailField |  |
| 5 | 管理员 | is\_staff | BooleanField |  |
| 6 | 超级管理员 | is\_superuser | BooleanField |  |

2、原始日志表设计，如表4-3所示：

**表4-3 原始日志表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 概念名称 | 字段名称 | 字段类型 | 宽度 |
| 1 | 日志ID | id | CharField | 40 |
| 2 | 系统类型 | systemtype | CharField | 20 |
| 3 | 接收时间 | gettime | DateTimeField |  |
| 4 | 等级 | classify | CharField | 20 |
| 5 | 源地址 | ipsrc | IPAddressField |  |
| 6 | 源端口 | srcport | IntegerField |  |
| 7 | 源用户名称 | srcname | CharField | 40 |
| 8 | 目的地址 | ipdst | IPAddressField |  |
| 9 | 目的端口 | dstport | IntegerField |  |
| 10 | 设备地址 | deviceads | IPAddressField |  |
| 11 | 设备类型 | devicetype | CharField | 40 |
| 12 | 时间名称 | event | CharField | 40 |

3、关联分析结果表设计，如表4-4所示：

**表4-4 关联分析结果表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 概念名称 | 字段名称 | 字段类型 | 宽度 |
| 1 | 告警日志名 | name | CharField | 40 |
| 2 | 最小支持度 | support | FloatField |  |
| 3 | 攻击序列 | attack\_sequence | CharField | 100 |
| 4 | 攻击名称 | attack\_time | CharField | 40 |
| 5 | 关联时间 | time | DateTimeField |  |

4、日志记录表设计，如表4-5所示：

**表4-5 日志记录表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 概念名称 | 字段名称 | 字段类型 | 宽度 |
| 1 | 记录名称 | name | CharField | 40 |
| 2 | 操作记录 | operation | CharField | 2 |
| 3 | 操作时间 | time | DateTimeField |  |

4.3 小结

本章将前一章设计的算法应用到实际系统开发当中，并对多源告警信息的关联分析系统进行了全面的剖析。并对各个子模块的功能进行了详细的解释，对数据库表进行了设计说明。针对某单位的告警日志进行了数据约定，实现了动态更新时间间隔阈值算法、关联分析算法、量化分析算法，同时也实现了各个模块的功能，进行了系统部署等任务。

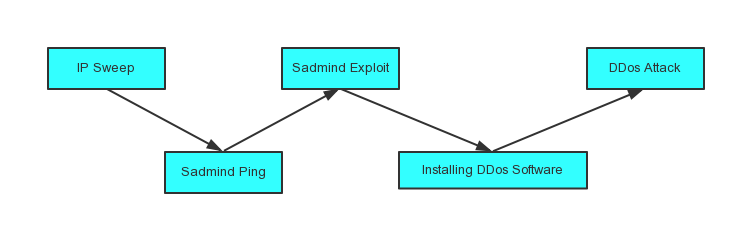
第5章 多源告警信息的关联分析系统评估及测试

5.1 实验测试数据

为了验证本文设计的算法的有效性，本文将采用darpa 2000(LLS\_DDOS\_1.0)数据集对算法进行验证，采用某单位（注：国家保密单位）提供的数据对算法和系统进行测试。

5.1.1 Darpa 2000数据集

darpa 2000是MIT Lincoln lab提出的数据集，该数据集是由模拟的入侵检测攻击的场景得出的，目前是比较权威的。图5-1给出了LLS\_DDOS\_1.0数据集五步攻击的过程。



**图5-1 LLS\_DDOS\_1.0数据集五步攻击过程**

Darpa 2000数据集提出的攻击分为五个阶段，每个阶段可以看作一个攻击。第一阶段是一个IP扫描(源IP 202.077.162.213)，会产生大量的请求响应包；第二阶段是对sadmind服务的验证；第三阶段是对sadmind缓冲区溢出攻击试着远程连接到主机；第四阶段远程连接跳板机并安装.rhost文件，安装mstream，控制守护程序的执行；第五阶段向目的IP(131.084.001.031)发起DDOS攻击。

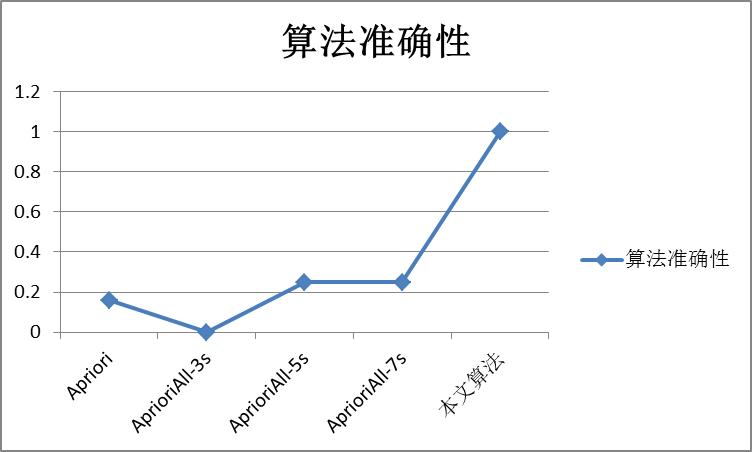
5.1.2 序列模式挖掘算法测试

根据Darpa 2000数据集提出的攻击场景测试本文设计的序列模式挖掘算法，关联结果如表5-1所示：

**表5-1 攻击步骤对应的关联分析结果**

|  |  |
| --- | --- |
| 攻击序列 | 关联度值 |
| ICMP PING ->ICMP Echo Reply | 0.82 |
| ICMP Echo Reply-> RPC portmap sadmind request UDP | 0.49 |
| RPC portmap sadmind request UDP-> RPC portmap Solaris sadmind port query udp request | 0.71 |
| RPC portmap Solaris sadmind port query udp request-> RPC portmap sadmind request UDP-> RPC sadmind query with root credentials attempt UDP | 0.91 |
| RPC sadmind query with root credentials attempt UDP-> RPC sadmind query with root credentials attempt TCP-> RPC sadmind UDP NETMGT\_PROC\_SERVICE CLIENT\_DOMAIN overflow attempt | 0.64 |
| RPC sadmind UDP NETMGT\_PROC\_SERVICE CLIENT\_DOMAIN overflow attempt-> RPC sadmind TCP NETMGT\_PROC\_SERVICE CLIENT\_DOMAIN  overflow attempt-> TELNET Solaris memeory mismanagement exploit attemp | 0.73 |

为了验证算法的有效性，运用对sadmind缓冲区溢出攻击试着远程连接到主机这个场景进行验证，将本文设计的算法与Apriori、AprioriAll算法进行比较，令最小支持度为0.3，AprioriAll算法的时间间隔阈值分别设为3s，5s，7s。通过现有的专家知识，来分析算法的准确性，具体分析情况如图5-2所示：



**图5-2 算法准确性比较**

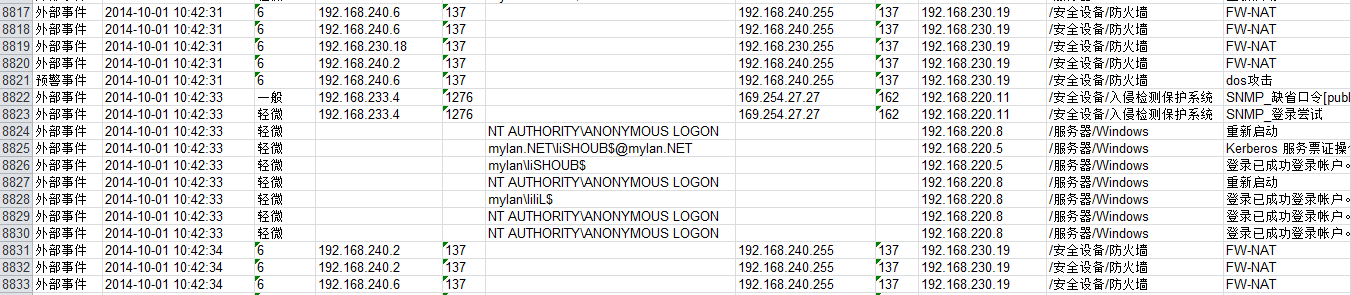
由图5-2所示，本文设计的算法略胜一筹。从图中可以看出，Apriori关联分析算法准确率较低，由此可以看出该算法并不适合于序列模式挖掘，它把各个攻击事件当作集合中的元素来对待，失去了序列原有的特性，从而降低了应用于序列模式挖掘的准确度。AprioriAll算法的时间间隔阈值需要一定的专家知识来判断，由上图所示，时间阈值影响这算法分析的准确性，选择适当的时间间隔阈值才能分析出更接近于正确的分析结果。本文设计的算法虽说准确度达到了100%，但并不表明测试任何数据的准确度都能达到100%，只能说提高了测试算法的准确度，该算法根据数据的特性，自动判断时间间隔阈值，使时间间隔阈值往更准确的方向变化，从而提高了算法一定的准确性。

5.1.3 某单位数据集

除了运用国际上比较有影响力的Darpa 2000数据集测试之外，还要利用某单位(注：国家保密单位)的网络环境下提供的真实数据进行详细的测试分析。由于本系统是针对于该单位的告警日志信息进行开发和设计的，因此运用该单位的数据集进行测试具有重大意义。由于该单位的保密性，攻击场景在这里就不予给出了。

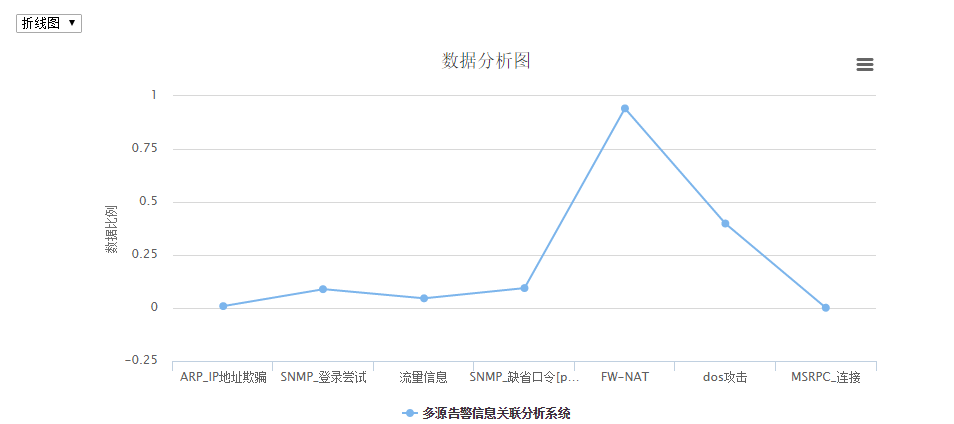
5.1.4 算法有效性测试方案及结果

图5-3为收集的原始告警日志信息。首先对告警日志信息进行预处理，过滤出IDS与防火墙的日志信息，提取出攻击事件类型以及攻击时间，然后根据攻击时间对攻击事件类型进行排序，然后再进行关联分析，得出攻击序列。



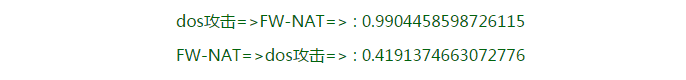
**图5-3 收集的部分原始日志**

在进行关联分析的同时会得出一个攻击统计图，表示在该时间段当中所发生的攻击以及各个攻击所占比例的大小。具体情况如图5-4所示：



**图5-4 攻击数据分析图**

关联分析算法结束后，会得出一个攻击序列以及关联度。关联度衡量着攻击序列的频繁程度。如图5-5所示：



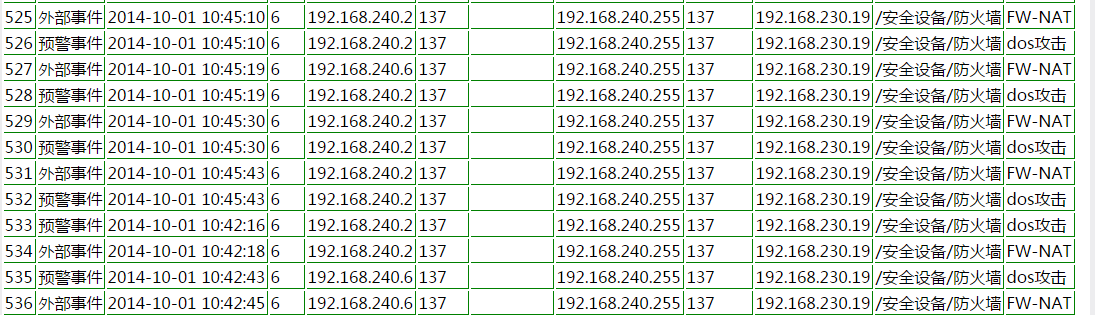
**图5-5 攻击序列及关联度**

得出攻击序列之后，会扫描与攻击序列有关的相关属性，从而更加直观的了解攻击序列中每条告警日志之间的关系，安全管理员可以更加了解该网络环境下的安全态势，方便管理员做出响应。如图5-6所示：



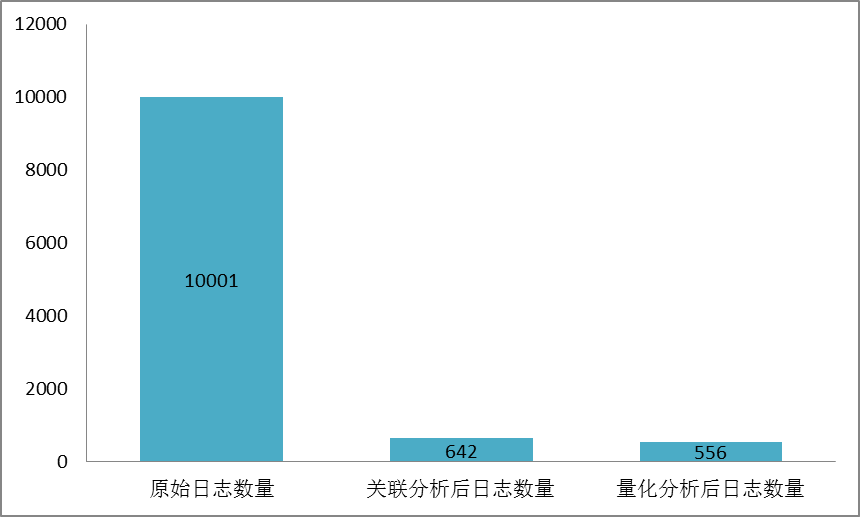
**图5-6 与攻击序列相关的部分告警日志信息**

为了得到更精准的数据，需要对上一步得出的数据进行量化分析，将小于最小分析度的告警信息过滤掉，从而得出符合要求的告警信息。最小分析度是要靠着先验知识来决定的，最小分析度设定的大小也影响着量化分析的结果，告警信息量化分析的具体结果如图5-7所示：



**图5-7 量化分析部分结果**

量化分析后，根据攻击序列得出的告警信息量有所精简。量化分析精简程度如图5-8所示：



**图5-8 日志关联分析量化分析结果**

本文设计的算法能够准确的挖掘出攻击序列并且应用于实际系统当中，高效的过滤掉冗余的日志，生成更精简的告警信息。算法能够自动的向用户提供推荐的最小支持度范围供用户参考，解决了对先验知识缺乏的安全管理员对最小支持度预算的问题。动态生成时间间隔阈值也解决了对时间间隔阈值把控不准的问题，提高了关联算法的准确性。量化分析对攻击序列的属性进行比较，从而可以得出更加精准的告警信息，可以应用于实时告警信息的监控。

5.2 系统功能测试

系统功能测试主要测试本文设计的系统的各项功能，测试浏览器为chrome。

5.2.1 用户管理模块测试

第6章 总结与展望

随着信息安全产业的不断壮大，信息安全产品的层出不穷，入侵检测系统、防火墙等边缘设备产生了海量的告警日志信息，仅靠专家知识很困难构建出完整的攻击序列，寻找攻击类型之间的关联迫在眉睫。本论文根据以上问题，提出了将动态更新时间间隔阈值的算法应用于本文改进的Apriori系列的序列模式挖掘算法当中，并且将生成的攻击序列的相关属性进行量化分析的方法。实验结果表明，这种方法可以挖掘出多步攻击模式的攻击事件，能够识别攻击者的攻击企图，重现攻击场景，并且具备量化分析的功能，提取出了更加有效的告警信息。

6.1 总结

本文对告警日志信息的关联分析和量化分析进行了较深入的研究，主要研究内容如下：

1、算法研究：本文研究了数据挖掘方面的Apriori算法，并将其应用到告警信息关联分析当中；研究了序列模式挖掘的AprioriAll算法并加以改进，将动态更新时间间隔阈值算法应用到序列模式挖掘算法当中，提高了算法的准确率。提出了一种量化分析算法，动态过滤不相关的信息。

2、系统实现：本文针对某单位(注：国家保密单位)的告警日志信息，运用python技术、django框架、前端技术对告警日志进行了关联分析、量化分析的结果呈现，形象且直观，减少了安全管理员对告警日志分析的复杂度。

3、测试评估：本文应用darpa 2000数据集和某单位(注：国家保密单位)数据集对算法和系统进行了测试。实验结果表明，该方法能有效的挖掘出攻击序列同时能对告警日志进行量化分析。

6.1 进一步工作

信息安全产业还在快速的发展，攻击方式越来越多样化，黑客的技术越来越高明，网络安全问题还是面临着巨大的挑战。由于本人水平有限，本文提出的方案还有许多地方值得深入的研究并且改进完善。本文的研究还属于初级阶段，离应用于产品阶段还有一定的距离，还需要做很多的研究工作，比如：

1、大数据方面。当数据量过大时，如何能实时的处理海量的告警数据。目前开发的系统处理10W以上的数据效率就会变低了，所以对大数据处理的方面有待研究。

2、量化分析算法的更新。本文采用的是基于属性之间的差别进行的量化分析，相对粗浅，需要在更加严谨的数学知识，来对量化分析算法进行描述。

3、如何通过得出的安全序列自动的分析出安全问题并根据下一步攻击的特征给予准确的防御，这是一个有待探索和研究的问题。

致谢

四年的大学生活转眼间就要结束了，真的可以用时光飞逝来形容我对这段时间的感受。虽然时间短暂，然而在这段时间里，不论是从学习上还是从生活上，我都学会了很多。这些收获都来源于帮助和关心过我的人，在这里我要向他们表示感谢。

本论文是在导师黄晓芳副教授的悉心指导和严格要求下完成的。从论文的选题、技术路线到论文撰写的整个过程中，黄老师都给予我精心指导和孜孜不倦的教诲。导师渊博的学识，严谨、求实、创新的治学精神，诲人不倦的师者风范，勤恳踏实的工作态度和乐观积极的人生态度深深地触动和影响了我，并将是我终身受益。在此，我要向我的导师黄晓芳教授表示衷心的感谢。

此外，我也要感谢大学里的各位同学给予我的帮助，感谢西南科技大学计算机科学与技术学院所有在本科阶段对我进行过教育和帮助的老师。

最后衷心地感谢我家人给予的支持、理解和无私的奉献，他们在背后默默的支持是我求学的动力。

参考文献

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases[C]. In Proc. of the ACM SIGMOD Conference on management of Data, Washington, D.C., May 1993: 207-216.

[2] Agrawal R, Srikant R. Fast algorithm for mining association rules[C]. In: Procedings of the 1994 International Conference on Very Large Data Bases, Santiago, Chile, 1994:487-499.

[3] J Han, J Pei,Y Yin. Mining Frequent Patterns Without Candidate Generation[C]. In Proc. of the 2000 ACM-SIGMOD Int.l Conf. on Management of Data, Dallas, Texas, USA,2000.1-12.

[4] 唐正军，李建华．入侵检测技术[M]．北京：清华大学出版社．2004．

[5] 张跃军.提高基于SVM的网络入侵检测性能的研究.山东大学.2006年4月.

[6] Valdes A, Skinner K. Probabilistic Alert Correlation. In: Proc. of the 4th International Symposium on Recent Advances in Intrusion Detection, Davis, California, USA, 2001. 54-68.

[7] Julisch K. Mining Alarm Clusters to Improve Alarm Handling Efficiency. In: Proceedings of the 17th Annual Computer Security Application Conference (ACSAC), December 2001.

[8] Staniford S, Hoagland J, Mcalerney J. Practical automated detection of stealthy portscans.Journal of Computer Security. Vol. 10, Issues 1/2, 2002.

[9] Templeton S, Levitt K. A Requires/Providers Model for Computer Attacks. In Proc. of the 2000 Workshop on New Security Paradigms, Cork, Ireland, 2001. 31-38.

[10] Ning P, Cui Y, Reeves D. Constructing Attack Scenarios through Correlation of Intrusion Alerts. In: Proc. of the 9th ACM Conference on Computer and Communications Security,Washington, DC, USA, 2002. 245-254.

[11] Ning P, Xu D. Learning Attack Strategies from Intrusion Alerts. In: Proc. of the 10th ACM Conference on Computer and Communications Security, Washington, DC, USA, 2003.200-209.

[12] Porras A, Phillip F, Martin W, Alfonso P. A Mission-Impact-Based Approach to INFOSEC Alarm Correlation. In: Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002), October 2002.

[13] Morin B, Me L, Debar H, Ducasse M. A Formal Data Model for IDS Alert Correlation. In:Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection(RAID 2002), October 2002.

[14] 王莉. 网络多步攻击识别方法研究: [博士学位论文]. 华中科技大学图书馆,2007. 5.

[15] 陶建明, 秦拯等. 一种基于数据挖掘的告警相关方法的研究与实现. 科学技术与工程, 2006(7): 65-73.

[16] 姚伟力, 王锡禄等. 基于序列模式挖掘的告警相关性分析算法. 北京邮电大学学报, 2005: 10.

[17] Malheiros MD. A model for alarm correlation in telecommunications networks[J].Belo Horizonte, 1997.

[18] Kimmo H, Klementtinen M, Mannila H, et al. TASA: telecommunication alarm sequence analyzer or how to enjoy faults in your network[A]. IEEE/IFTP, 1996 Network Operations and Management Symposium (NOMS’96[C]. Kyoto, Japan,1996: 520-529.

[19] 毛国君, 段立娟等. 数据挖掘原理与算法. 北京: 清华大学出版社, 2005.

[20] Ian H. Witten, Eide Frank. Data Mining: Practical Learning Tools and Techniques with Java Implementations. Elsevier, 2003.

[21] 邓维斌,王国胤,王燕.基于RoughSet的加权朴素贝叶斯分类算法[J8计算机科学,2007,2.

[22] M. H. Kang and T. Mayfield. A cyber-event correlation framework and metrics.Proceedings of the SPIE - The International Society for Optical Engineering,2003(5107): 72.

[23] R. P. Goldman, W. Heimerdinger, S. A. Harp, C. W. Geib, V. Thomas, and R. L.Carter. Information modeling for intrusion report aggregation. Presented at DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings, 2001.

[24] J. Yu, Y. V. R. Reddy, S. Sentil, K. Srinivas, R. Sumitra, and B. Vijayanand.TRINETR: an intrusion detection alert management systems. Presented at Proceedings. Thirteenth IEEE International Workshops on Enabling Technologies:Infrastructure for Collaborative Enterprises, Modena, Italy, 14-16 June 2004.

[25] 周杰.网络攻击预警系统的研究、设计与实现.[硕士学位论文].北京邮电大学 2010.

[26] 张连华.基于层次的智能告警关联分析模型研究.微型电脑应用，2011年8期.

[27] 王豪.多源监控告警数据相关性分析的研究与应用.[硕士学位论文].电子科技大学 2014.