孙立钢-防火墙日志分析系统的设计与实现-15.23%

【PDF报告-大学生版】

报告编号：ccd01a5efdcfa93a　　检测时间：2018-04-16 08:04:36　　检测字数：19,963字

作者名称：孙立钢　　所属单位：杭州师范大学（教务处）

检测范围：

◎ 中文科技期刊论文全文数据库　◎ 中文主要报纸全文数据库　◎ 中国专利特色数据库
◎ 博士/硕士学位论文全文数据库　◎ 中国主要会议论文特色数据库　◎ 港澳台文献资源
◎ 外文特色文献数据全库　◎ 维普优先出版论文全文数据库　◎ 互联网数据资源/互联网文档资源
◎ 高校自建资源库　◎ 图书资源　◎ 古籍文献资源
◎ 个人自建资源库　◎ 年鉴资源　◎ IPUB原创作品

时间范围：1989-01-01至2018-04-16

检测结论：

全文总相似比：10.70%　（总相似比=复写率+他引率+自引率）
自写率：89.30%　（原创内容占全文的比重）
复写率：9.62%　（相似或疑似重复内容占全文的比重,含专业用语）
他引率：1.08%　（引用他人的部分占全文的比重，请正确标注引用）
自引率：0%　（引用自己已发表部分占全文的比重，请正确标注引用）
专业用语：0.00%　（公式定理、法律条文、行业用语等占全文的比重）

总相似片段：63

期刊：16 博硕：22 外文：0 综合：0 自建库：0 互联网：25

本 科 生 毕 业 论 文 正 文

（ 2018 届）

论文题目 防火墙日志分析系统的设计与实现

学生姓名 孙立钢 学 号 2014211536

专 业 计算机科学与技术 班 级 计算机143

指导教师 刘雪娇 职 称 副教授

杭州国际服务工程学院教学部

防火墙日志分析系统的设计与实现

摘 要

防火墙技术已经广泛应用于企业服务器和个人计算机，作为内部网络和外部网络之间的安全系统，可以有效的将威胁隔离在外部网络，保护计算机处于相对安全的状态下。防火墙日志记录系统能够有效记录发生在当前网络中的网络流量。然而防火墙日志数据量大且离散，用户难以直接发现引起网络问题的日志，因此有必要对防火墙日志进行分析，展现网络状态和发现网络异常的原因。

本文通过分析海量的防火墙日志，设计了防火墙日志分析系统，采用统计分析和关联分析的方法，达到发现网络异常，帮助用户辅助分析网络状态。该系统由分析端和展示端组成，其中分析端对日志进行相关分析，而展示端则提供了RESTFul API，并在前端页面中展示分析结果。

关键词：日志采集，统计分析，关联规则，防火墙日志

Design and Implementation of Firewall Log Analysis System

ABSTRACT

Firewall technology has been widely used in enterprise servers and personal computers as a security system between internal networks and external networks. It can effectively isolate threats from external networks and protect computers in a relatively safe state. The firewall logging system can effectively record network traffic occurring in the current network. However, the log data volume of the firewall is large and discrete. It is difficult for the user to directly discover logs that cause network problems. Therefore, it is necessary to analyze the firewall logs to show the network status and the cause of the discovery of the network exception.

This paper analyzes a large number of firewall logs and designs a firewall log analysis system. It uses statistical analysis and correlation analysis methods to achieve the purpose of discovering network anomalies and providing user assistance in analyzing network status. The system consists of an analysis end and an exhibit end. The analysis end analyzes the logs, and the presentation end provides the RESTFul API and displays the analysis results on the front

end page.

Keywords: Log collection, statistical analysis, association rules, firewall logs

目 录

第一章 绪论

本章主要介绍防火墙日志分析系统的研究目标和研究意义，描述了研究内容和研究目标。

1.1 课题研究及研究意义

防火墙作为网络安全的重要防线，不论是个人电脑还是企业服务器都广泛使用了防火墙技术，对于防御这些网络攻击有巨大的作用。同时对防火墙日志进行相关分析能够为提高网络安全状况提供帮助，从而进一步提高网络的稳定性。在分析的过程种我们使用到了关联规则挖掘算法，关联规则挖掘算法能够有效发现日志之间的联系，从而分析出可能存在的网络安全事件。

1.1.1 防火墙技术研究

当前的防火墙系统主要分为应用层防火墙和网络层防火墙。应用层防火墙包括FTP服务器、Web服务器和邮箱服务器等流量，该层次防火墙能够过滤指定应用程序的所有封包（依据一定通信协议的完整数据包传输过程）。网络层防火墙主要针对数据包的源目的IP、源目的端口、所用协议等进行过滤。分别属于代理防火墙和包过滤防火墙，其中代理防火墙又称应用层网关防火墙。

在防火墙原理层面上，代理防火墙采用了网络代理，对防火墙内部的网络IP地址和网络拓扑进行隐藏和保护，使得外部网络无法探知防火墙内部真实的网络拓扑和具体IP对应机器的服务器功能，从而达到最大程度保护内部网络的目的。而包过滤技术又分为静态包过滤技术和动态包过滤技术，其中静态包过滤技术是第一代过滤技术，采用简单配置和匹配数据包头部信息，虽然过滤快速，但是无法考虑多个数据包之间的联系和深入数据包内部，检测负载的信息。动态过滤技术作为第二代包过滤技术，是在第一代静态包过滤技术的基础上发展起来的，是基于多个数据包进行状态分析的包过滤技术，与静态包过滤技术相比，没有速度上的优势，但是能够探测多个数据包之间的联系并探测数据包负载的信息，从而提高了防火墙系统的可靠性。

Iptables作为常用的Linux包过滤器，前身叫ipfirewall，能够工作在Linux内核当中的，对数据包进行检测的一款简易的访问控制工具。而发展到现在iptables已经具有精细的访问控制能力。Iptables本身并不是实际意义上的防火墙，而是作为定义规则的工具，真正实现过滤功能的是netfilter（网络过滤器）。Iptables分为四个表，分别是Iptables 有 四个表 Raw表、Manage表、Nat表、Filter表，其中Filter是默认表。四个表的优先级分别是Raw > Manage > Nat > Filter。同时拥有5个链，分别是PREROUTING、ROUTING 、FORWARD 、INPUT 、OUTPUT 、POSTROUTING。当数据包到达网卡时会根据不同的链和表的配置，进行不同的数据包路由。

1.1.2 日志系统研究

Linux 系统日志对运维人员具有极其重要的作用，当运行中的服务器出现问题的时候，问题排查第一步是查看日志信息。同样，对于网络管理员，日志信息同样具有重要的价值。Syslog作为常用的Linux日志系统，该服务由klogd和syslogd两个进程完成。其中klogd来记录内核日志，一般保存系统初始化日志，产生的日志文件在/var/log/dmesg ；syslogd进程来记录系统日志，一般保存非内核日志，产生的日志文件在/var/log/messages。而Linux防火墙的日志由syslogd产生并保存在/var/log/messages。想要对其进行相关分析，需要进行必要的日志清洗和提取有效字段，目前常用的手段有采用正则表达式匹配和日志SQL等，正则 表达式通过分析日志的特征书写响应的正则表达式并匹配每一条日志来提过有效信息，而日志SQL化将日志列表看做一张数据表，并通过分割符来区分各个字段。

1.1.3 防火墙日志的作用

日志作为硬件或软件设备保持在文件系统或者数据库中的信息，具有较大的价值。日志的来源决定了日志的作用。例如入侵检测系统的日志记录了网络异常而发出的警告信息，而防火墙日志则是对于网络数据包的不同行为产生的信息。总之，日志告诉了用户，软硬件设备发生的不同行为。通过日志，用户可以了解系统的运行状况，安全状况。在当前网络安全事件频繁发生的状态下，分析防火墙日志有现实意义。例如当发生一次拒绝服务攻击时，攻击者会扫描被攻击网段，希望找出漏洞，因此表现在防火墙日志上的特征就是同一源IP在短时间内访问了同一个目的IP下的多个端口或者是同一网段下的多个IP。当攻击者发现网站或者服务器的漏洞时，则可以发起拒绝服务攻击，在防火墙日志上的表现就是多个源IP在短时间内大量访问目标IP，造成服务器资源的耗竭，形成拒绝服务。

1.1.4 关系规则挖掘算法在网络安全中的意义

关联规则表现在网络安全中，更多体现是安全事件之间的联系，也可以是日志之间的联系。Apriori算法作为一种经典的关联规则算法能够有效得挖掘网络安全事件之间的关联规则。包括防火墙日志在内的网络安全日志都存在数据量大的特点，往往以每天TB级别的数据量出现，要手动分析如此海量的日志不现实的，因此利用关联规则挖掘算法能够找到海量数据之间的联系，便于用户进行后续网络攻击的发现。同时利用关联规则的置信度还可以发现安全事件的先后联系，例如事件A发生了事件B总是发生，这样可以发现更加强烈的关联规则，便于分析攻击的行为方式，通过不断总结可以形成网络攻击知识库，便于预防相关攻击。关联规则在网络安全中有巨大的作用，用在防火墙日志分析领域同样也能够发现目的IP之间的关联关系，可以用来分析用户的行为方式，比如用户在访问web页面之前会进行DNS解析等用户行为分析。

1.2 研究内容与目标

实现一个即具有实时日志数据分析功能同时又具有历史数据分析的防火墙日志分析系统。在使用过程中将分析端和展示端相互分离，其中展示端提供RESTful API。并且使用关联规则挖掘算法，挖掘防火墙日志之间可能存在的关联关系。

分析端和展示端的相互独立，在降低耦合度的同时，也能够保证分析端分配到足够的CPU时间，保证其计算效率。

采用RESTful API，其良好的透明性并且充分利用 HTTP 协议本身语义，最重要的是无状态，不需要考虑应用上下文，能够保证前后端的低耦合。

使用关联分析算法，关联分析算法能够有效发现防火墙日志之间的联系，例如IPA和IPB总是同时出现，那么这两个IP的关联度较高。

1.3 本章小结

该章节主要介绍了课题背景和与研究内容与目标，包括防火墙技术、日志系统、关联规则算法等，同时明确了研究目标是实现一个即具有实时日志数据分析功能同时又具有历史数据分析的防火墙日志分析系统。

第二章 相关技术

该系统在日志分析方面，使用了日志解析技术、关联规则挖掘算法。而在系统构建和开发方面使用了常用的Java企业级框架，包括Spring boot、Mybatis、Hibernate等，关系型数据库MySQL数据库和可视化控件百度Echarts等。

2.1 日志分析相关技术

2.1.1 日志解析技术

正则表达式常用于日志解析工作，其具有效率高，匹配速度快的特点。正则表达式算法通常采用有限自动机用来表示一组特征正则表

达式[2]。同时正则表达式提供了用户根据自定义的匹配规则来解析字符串的功能，能够灵活地分割和提取有效字段。使用正则表达式能够方便快捷地提取防火墙日志的各个关键字段，尤其是针对日志格式复杂，且没有明显的限定的分割符，比如空格等分隔符。在当前日志格式下使用正则表达式能够在保证解析效率的前提下，快速有效得完成字符串解析工作。

2.1.2 Apriori算法

Apriori[4]是一种经典的数据挖掘算法，利用迭代方法，不断生成更多项的频繁结合，整个过程需要多次扫描数据结合，最终得到关联规则。在Apriori算法中有两个比较重要的概念，一个是支持度，另一个是置信度。支持度指前项和其他项在同一个频繁集合中存在的概率；而置信度则是一个条件概率，即前项发生时，其他项的发生的概率，用来保证频繁结合的可信度。

虽然是一项经典的关联规则挖掘算法，但是由于其算法设计的问题，需要多次扫描数据集合，在面对大数据量的情况下，常常需要消耗大量时间，因此提出了改进版的Apriori算法和基于决策树减枝的关联规则挖掘算法，后者在性能上有了较大的提升，最好的情况下效率提升了2倍以上[5]，但是算法实现的复杂程度也随之增加了。

2.2 系统开发相关技术

2.2.1 Spring介绍

Spring框架是一个分层架构，由7个定义良好的模块组成。Spring由7大模块组成，分别是核心容器、Spring 上下文、Spring AOP、Spring DAO、Spring ORM、Spring Web 模块、Spring MVC 框架。其框架图如下：

图2-1 Spring 模块架构图

2.2.2 Spring boot 介绍

Spring框架虽然提供了良好的控制反转和面向切面编程的方案，由于Java Web开发使用到的技术繁杂，使Spring的配置工作十分复杂，给初学者造成较大的困难。随着开源社区的反馈，Pivotal推出了Spring boot。

Spring Boot的颠覆性创新在于：在框架内部嵌入Jetty、Tomcat等作为Servlet container。不需要将项目打成war包后发布到application server里面，而是像运行java本地程序一样运行web项目；提供了Spring boot starter POM,极大简化了包管理方式；根据导入的依赖，自动进行Spring框架的配置，使程序员专注在业务逻辑，而不是繁琐的配置问题；不需要任何的第三方系统,Spring Boot实现了适用于多种生产环境的程序状态信息和健康状态机制，同时可以让应用程序非常方便的读取外部的配置信息；完全不依赖任何自动生成的代码，也不需要通过xml配置文件来进行框架的配置和管理。因此，Spring boot解决了一系列的开发痛点，包括从项目开发前的配置、项目开发中的依赖管理、项目开发完成后的部署等。

2.2.3 MySQL数据库介绍

MySQL数据库是一个常用的开源关系型数据库，是当前主流的关系型数据库，又由于其开源免费的优势，被大量使用于教学和中小型应用。MySQL提供了规范的SQL语言支持，并且有大量大学课程以MySQL为例，使得MySQL的学习成本大幅度下降，因此在本系统中使用MySQL数据库作为存储系统是一个经济且恰当的选择。

同时，关系型数据库从诞生到现在经过几十年的发展，已经变的比较成熟，目前市场上主流的数据库都为关系型数据库，比较知名的如 Sybase，Oracle，Informix，SQL Server，DB2 等-[3]。MySQL作为经典的关系型数据库，有容易理解、使用方便和易于维护等优势。更重要的是关系型数据库相较于NoSQL数据库，一旦遵循良好的范式来进行设计能够大幅度降低数据冗余，针对存储资源有限的服务器和个人计算机有着先天的优势。

2.2.4 Mybatis和Hibernate介绍

Mybatis和Hibernate是当前Java后端开发中使用最多的两款ORM框架，所谓的ORM框架是Object Relational Mapping,即对象关系映射。ORM框架解决了在开发过程中SQL书写繁杂且重复的问题，将面向过程的SQL和数据库连接等操作抽象为面向对象的形式，使得整个软件开发流程更加面向对象，更为便捷。Mybatis和Hibernate都占有着较大的市场，各有各的优势，在分析日志时，因为需要更多的多表操作，我们使用更好支持了多表操作的Hibernate；而在前端获取数据时，我们使用了能够更加方便操作SQL的Mybatis。

2.2.5 Echarts和C3.js介绍

在前端开发中，我们经常需要使用图表用来展示数据，这种情况下我们就经常需要使用第三方的图表库进行二次开发。百度Echarts是国内领先的基于JavaScript图表库，支持常用的柱形图、饼图、折线图等，同时可以扩展支持数十种以上的新型图表。

而C3.js则是在D3.js的基础上发展起来的图表软件，相较于Echarts，D3.js更加轻型占用系统资源也更加节省，因此在需要绘制动态图表的情况下，可以考虑采用C3.js。

2.3 本章小结

本章介绍了该系统所使用的主要的工具、技术和相关算法。包括日志解析使用的正则表达式，系统框架使用的Spring boot、Spring、Mybatis和hibernate等，前端展示时使用的Echarts、C3.js 和bootstrap等技术。

第三章 系统设计

本章介绍了系统的设计，包括分析端和展示端，已经分析端使用的分析方法和展示端使用的架构。

3.1 系统架构设计

系统分为两个子系统，一个是分析端，另一个是展示端。

系统架构图如下：

事件判定

实时分析

关联分析

分析层

日志采集

日志监听

日志解析

日志预处理

采集层

事件判定结果

实时分析结果

关联分析结果

展示层

分析结果记录

图3-1防火墙日志分析系统架构图

采集层负责日志的监听、采集、解析和预处理。当有新日志产生时，能够及时读取日志并进行解析并持久化到数据库。

分析层负责日志的分析，分为实时分析、事件判定和关联分析。实时分析进行流量计算，各个字段的统计。事件判定是发现网络的异常行为，包括同源大量访问、同目的大量访问和访问敏感端口。而关联规则则用来发现用户的行为方式。展示层用来展示分析结果和记录安全日志。

3.2 分析端设计

分析端有两块功能，分别是实时数据分析和历史数据分析。

在进行分析之前需要进行日志采集，日志采集模块的工作是，监听日志文件的变化，一旦有新的日志产生时，需要及时读取日志并进行分析。读取到的日志先进行正则表达式匹配，解析出包括源IP、目的IP、源端口、目的端口等在内的字段，并将数据及时持久化到数据库中。

实时数据分析，是针对实时产生的防火墙日志进行分析，主要采用统计方式。包括统计短时间内的数据流量和检测日志访问的端口，比如是否访问敏感端口。敏感端口是指一些在已经注册了特殊功能的端口，访问这些端口往往需要特殊的权限。同时统计一定时间内的所有源IP的访问次数，所有源端口的访问次数、所有目的IP的访问次数、所有目的端口的访问次数，用以评估当前的网络状态。将

网络事件分为三种，分别是同源大量访问事件、同目的大量访问事件和访问敏感端口访问事件。其中同源大量访问事件为同一个IP在短时间内大量访问当前网络；同一个目的IP被大量访问是指同一个目的地址在短时间内被大量访问；敏感端口访问是指如23端口这样的敏感端口被访问。

表3-1 网络事件分类

事件名称 描述

同源大量访问事件同目的大量访问事件敏感端口访问事件 同一个IP在短时间内大量访问当前网络.可能是DOS攻击同一个目的IP在短时间内被大量访问，可能是Scan攻击敏感端口被访问，根据端口不同，存在多种攻击可能

历史数据分析，针对较长时间的历史数据进行分析，主要包括较长时间的历史数据统计，这部分统计的方法与实时数据统计相同，只是在时间范围上选取更大，以便于给出更加宏观的网络访问情况。历史数据的关联规则挖掘，使用Apriori算法进行关联规则挖掘，希望通过Apriori算法挖掘出一定时间内的源IP的访问规律，希望探查哪些IP经常一起出现。用在防火墙日志分析领域同样也能够发现目的IP之间的关联关系，可以用来分析用户的行为方式，比如用户在访问web页面之前会进行DNS解析等用户行为分析。。频繁项集合计算示意图如下：

图3-2 频繁集合计算示意图

下表是分析端的主要功能：

表3-2 分析端功能

功能模块 详细

日志采集实时统计安全事件判定关联规则挖掘 日志监听、日志采集、日志解析源目的IP、端口统计，实时流量计算、敏感端口探查源目的IP、端口统计，发现异常统计目的IP之间的关联规则，发现用户的行为方式

3.2.1 日志采集

日志采集分为日志监听和日志解析两个部分，其中日志监听的作用是监听是否有新日志产生，而日志解析则是将文本日志的各个字段解析出来并持久化到数据库中，其流程如图3-3所示。其中日志监听流程图如图3-4所示。

在监听到新的日志产生后需要及时对日志通过正则表达式进行解析。在使用正则表达式进行解析时，需要对不同字段书写不同的正则表达式以提取对应的字段。相关字段与正则的对应关系如表3-3所示。

图3-3数据采集流程图 图3-4 日志监听流程图

表3-3 正则表达式和日志相关字段对应关系

字段名称 正则表达式

内网IP源IP和端口

目的IP和端口时间戳协议号

(\d{1,3}\.){3}\d{1,3}(\d{1,3}\.){3}\d{1,3}\(\d+\)->->(\d{1,3}\.){3}\d{1,3}\(\d+\)\d{4}(-\d{2}){2}: (\d{2}:){2}\d{2}

协议：\d{1,3}，

3.2.2 实时统计

在实时统计的过程中，对一定时间内的日志对于源IP、目的IP、源端口、目的端口进行统计并保存到数据库中。其流程如下图所示。

图3-5实时统计流程图

日志统计方法较为简单，主要是使用MySQL的SQL语句进行统计算法，可以最大限度减少代码的书写，同时提高统计的时间效率，降低复杂度，满足对于实时性的需求。

3.2.3 安全事件判定

安全事件判定需要一定的安全知识，在DOS攻击中，表现在防火墙的日志上，就是大量同源IP的同时访问，以造成服务器的无法及时应答。而攻击之前的必要的扫描，则表现在防火墙日志的大量同源不同目的的日志上。因此通过分析多条防火墙日志的关系可以找到网络异常，甚至分析出网络攻击。

网络安全事件的判定，需要维护一个事件集合，这些事件并不一定最后形成完整的事件，每当有一条新日志进入时，需要根据所有的事件类型遍历匹配所有的可能的事件，当某个事件的开始时间和当前日志的结束时间大于阈值时，则判定该事件是否形成某个类型的安全事件，一定形成则保存到数据库中。安全事件判定的流程图如图3-6所示。

3.2.4 日志关联分析

日志的关联分析用到了Apriori算法，这个经典的关联规则挖掘算法。算法通过不断迭代的形式不断挖掘频繁集合。并通过计算条件概率，即置信度的形式来保证规则的可行度。其大致流程如下图3-7所示。

图3-6 安全事件判定流程图 图3-7 关联分析流程图

3.3 展示端设计

展示端主要将分析端的结果可视化地展示给用户，以便于用户更加形象化地感受网络状态的变化，以便于做出较为准确的防火墙设置。使用Echarts和C3.js提供的图表库，我们可以将实时统计结果和历史分析结果形象地展示在前端页面上。

同时为了便于用户记录每天的时间，添加了日记记录功能。

下表是展示端的主要功能：

表3-4 展示端功能

功能模块 详细

登录模块实时统计展示模块历史统计展示模块事件展示模块日志模块 用户登录实时流量展示，统计结果展示

历史统计结果展示，关联规则分析结果展示展示发生的安全事件

日志编写、上次、修改、分页

3.4 数据库设计

用户表，用以记录用户相关信息。

表3-5 用户表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idname

password

phoneNumber BigIntVarchar(20)Varchar(20)Varchar(20) 无

无

无

无 是

否

否

否 否

否

否

否 ID字段用户名用户密码电话号码

日志表，用以保存解析出来的日志的各个字段。

表3-6 日志表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idinternalIptimestamporiginalSrcIporiginalSrcPort originalDestIPoriginalDestPort convertedSrcIp

convertedSrcPortconvertedDestIPconvertedDestPortprotocolNumbersafefymarginaciton BigIntVarchar(20)

DatetimeVarchar(20)

Varchar(5)

Varchar(20)

Varchar(5)

Varchar(20)

Varchar(5)

Varchar(20)

Varchar(5)

Varchar(5)

Varcher(10)Varcher(4) 无

无

无

无无无无无无无无无

无 是

否

否

否否否否否否否否否

否 否

否

否

否否否否否否否否否

否 ID字段内部IP时间戳原始源IP地址原始源端口原始目的IP原始目的端口转换后源IP转换后源端口转换后目的IP转换后目的端口协议号数据包流向

防护墙行为

事件表，用来保存计算出来的网络事件。

表3-7 事件表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idstartTime

endTime

isFinishedtype BigInt

datetime

datetime

tinyInt(1)enum 无

无

无

无无 是

否

否

否否 否

否

否

否否 ID字段事件开始时间事件结束时间

事件是否完整

时间类型

日志和事件关系表，由于日志和事件是一对多的关系，因此从范式的角度，需要独立出新的关系表来保存两者之间一对多的关系。

表3-8 关系表

字段名称 数据类型 默认值 是否主键 是否外键 说明

fwlogId

eventId BigInt

BigInt 无

无 否

否 是

是 日志ID事件ID

统计结果表，保存一段时间内的统计结果。

表3-9 统计结果表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idstartTime

endTimestatisticsValuecounttype

abnormal BigInt

datetime

datetime

varchar(20)intenumtinyint 无

无

无

无无无是 是

否

否

否否否否 否

否

否

否否否否 ID字段本次统计开始时间本次统计结束时间

统计对象值

次数统计类型是否正常

敏感端口表，保存所有的敏感端口。

表3-10 敏感端口表

字段名称 数据类型 默认值 是否主键 是否外键 说明

portdesc Varchar(5)Varchar(255) 无

无 是

否 否

否 敏感端口描述

访问流量表，保存计算出来的访问流量。

表3-11 访问流量表

字段名称 数据类型 默认值 是否主键 是否外键 说明

identTimecount BigIntDatatime

Int 无

无无 是

否否 否

否否 ID字段统计结束时间流量大小

关联规则结果表，保存计算出来的关联规则。

表3-12 关联规则结果表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idstartTime

endTime

collection BigInt

datetime

datetime

varchar(255) 无

无

无

无 是

否

否

否 否

否

否

否 ID字段事件开始时间事件结束时间

关联规则结果

日志记录表，保存每天记录的日志，以便于后续查阅当天的情况。

表3-13 日志记录表

字段名称 数据类型 默认值 是否主键 是否外键 说明

idcreateTime

titlerecordInfouserId BigInt

datetime

varchar(20)

textBigInt 无

无

无

无无 是

否

否

否否 否

否

否

否是 ID字段创建时间

记录标题

记录内容创建者ID

3.5 本章小结

本章主要介绍了系统的总体设计，并分开介绍了分析端和展示端的设计。然后详细介绍了数据库表的设计。

第四章 系统实现

本章介绍完整系统的实现和运行效果和相关测试效果。

4.1 系统环境

本系统主要采用Java语言开发，使用JDK版本为1.8版本，是目前主要使用的JDK版本，Java语言具有良好的跨平台能力，方便后期部署在Linux服务器上。同时开发平台使用的是windows10系统和MySQL数据库。下表是详细情况：

表4-1 开发环境配置

配置名称 版本 备注

Windows系统 10 操作系统

内存 8G 满足基本开发需求

JDK 1.8 版本为1.6以上即可

IntelliJ IDEAGradle

GitMySQL 2017.34.02.10.0.windows.15.7.13 主流开发IDE构建项目项目管理数据存储

4.1.1 分析端结构

分析端使用Gradle工具构建代码，项目使用Spring boot快速生成，在数据层使用Hibernate进行对象关系映射，使用Spring管理对象实例。采用层级结构分别为entity-dao-analysis。

在使用Gradle构建项目时，不需要手动添加依赖Jar包，只是需要添加相关依赖的名称，就会自动添加相关需要使用的java包。由于Gradle使用的是Maven的Jar包库，下载速度存在较大的问题，因此使用阿里巴巴的代理。分析端添加的依赖如表4-2所示。

表4-2 分析端依赖

依赖 说明

compile "org.hibernate:hibernate-core:5.0.0.Final"compile group: 'com.google.code.gson', name: 'gson', version: '2.8.0'compile group: 'log4j', name: 'log4j', version: '1.2.17'runtime('mysql:mysql-connector-java')testCompile('org.springframework.boot:spring-boot-starter-test')

Hibernate相关功能提供Json功能日志功能连接MySQLSpring boot测试功能

4.1.2 展示端结构

展示端同样使用Gradle 进行项目构建，在层次结构上使用典型的MVC架构。MVC是(model)一视图(view)一控制器(controller)的缩写。分层的思想简化了各层的功能，易于梳理整个项目的逻辑。其中 Model是应用的关键实现各项业务，View用来展现数据并进行交互，Controller则提供了相关接口用来数据交互。MVC模型如图4-1 所示。

图4-1 MVC模型图

根据MVC的架构，展示端的结构比分析端多了Controller层和View层，并且需要增加更多的依赖包，以便提供web服务。

在页面展示方面，使用Bootstrap搭建主题框架，Bootstrap是Twitter公司推出的一款前端框架，具有良好的响应式设计，能够完美得自适应各种设备，因此广泛应用于前端开发。并且由于遵循了同一套CSS，在代码风格上实现了统一，有利于合作开发。

同时，为了方便前后端的分离开发，将展示端的后端和前端独立为两部分，但是这涉及到了跨域的问题。域（DOMAIN），作为一种安全域界，是Internet的逻辑组织单元[6]。所谓跨域问题，就是请求发起者和资源提供者并不在同一个域中。解决跨域问题，常用的方法有两种，第一种是使用ajax的JSONP，JSONP作为一种支持多种浏览器内部跨域的信息交换技术，常用于不同域间数据的传递，该技术由于灵活方便的特性，使JSONP在Web应用中得到了广泛的使用。虽然JSONP使用方便且灵活，但是其存在较大的安全问题[7]。另一种方法是CORS，全称是跨域资源共享（Cross-Origin Resource Sharing）。它允许浏览器向跨源服务器，发出XMLHttpRequest请求，从而克服了AJAX只能同源使用的限制。CORS的使用需要服务器和浏览器都支持。目前，绝大多数的浏览器厂商都实现了该功能，实现整个CORS通信过程，不需要用户参与，而由游览器自动完成。这简化了开发人员的开发流程，和JSONP相比，不需要额外的前端代码。当浏览器发现了AJAX需要请求跨域资源，自动添加跨域相关的头信息，并发送相关请求。在服务器端需要支持CORS服务，在Spring boot中开启CORS的方法较为简单，使用WebMvcConfigurer接口和CorsRegistry类重新映射原始请求。

4.2 分析端日志采集

日志采集分为两个功能模块，分别是日志监听和日志解析。日志监听实时监听日志文件并及时读取日志。日志解析是使用正则表达式解析日志得到相关日志字段。

4.2.1 日志监听

日志监听容易实现的方法就是监听的日志文本的长度。当日志文本的长度发生变化时，说明有新的日志产生需要及时进行读取。同时为了方便管理全局变量，在consts包下建立ConfigConsts接口，用来保存全局变量，包括配置文件的位置。

4.2.2 日志解析

当前对于文本日志的解析工作，使用正则表达式进行解析。下面给出原始日志数据的例子，

Aug 1 00:00:00 10.136.14.155 2018-03-27: 14:45:43 FWinternetDMZMain:root 03-01-069-0018 Notice Session N/A rep=1 | 匹配到访问策略Deny远程连接，原始地址：123.123.12.12(35257)->101.67.162.33(23)，协议：转换后地址：81.0.24.102(35257)->101.67.162.33(23)， 安全域：outside->server， 动作：阻断。 针对该格式的日志，我们需要分别解析出时间戳（timestamp）、内网IP（inter）、转换前源IP（original source IP）、转换前源端口（original source port）、转换前目的IP（original destination IP）、转换前目的端口（original destination port）、转换后源IP（converted source IP）、转换后源端口（converted source port）、转换后目的IP（converted destination IP）、转换后目的端口（converted destination port）、协议号（protocol number）和访问域（access domain）。

由于正则表达式匹配时，可能存在匹配失败的问题，因此引入解析失败异常，在解析失败时，处理该异常。

4.3 分析端日志分析

4.3.1 日志流量分析

由于网络事件发生在短时间内，在进行实时统计时，选取的时间间隔不需要特别长，根据测试的结果，选取为2秒钟。统计了每个两秒钟内的日志条数，来计算出日志的流量大小。针对日志流量，抽象出AccessFlowResult类，使其与数据库表中的AccessFlowResult表相互映射，映射的过程使用了Hibernate的注解Entity，并使用Column(name = "字段名称") 注解相关字段，并保证name值需要和数据库表中的字段的值相同。直到当前日志的时间戳和上次统计的日志的时间戳间隔大于预设的时间间隔（1秒钟）时，该次流量统计结束，并将结果插入数据库中。

4.3.2 日志统计分析

对于不同字段的统计值，其实是统计方法是相同的，只是统计的字段的不同，于是使用Java的反射机制。Java的反射机制作为Java的一种重要特性在Java开发中具有重要的应用。在几乎所有的ORM框架中都使用了反射技术[8]，Java 语言内置了功能十分强大的反射方法，使程序在其运行期间也可以获取元数据并且实现类的动态加载，并且最终生成实例对象，列出类有哪些方法和字段，并通过invoke方法调用。使用Java的反射机制可以大大简化代码的书写，尤其是减少重复代码。

在统计时，当计入一条日志信息时，首先判断在数据库中是否已经有了该次统计的统计结果，如果有则只需要进行更新，否则需要新建一条记录用来记录这次统计。将不同类型的统计结果保存到统计结果表中，用于前端展现。

4.3.3 安全事件判定

将同源大量访问，同目的大量被访问和访问敏感端口设置为安全事件。对日志进行分析，计算出相关的安全事件，并将安全事件与日志的关系记录下来，便于后续的继续分析。对于同源大量访问和同目的大量访问，计算方式是相同的。首先需要维护一个事件集合，当新读入一条日志，则遍历一次该事件集合，一旦某个事件的开始时间和该日志的时间戳的时间间隔大于一定的阈值时，先判断该事件是否构成一个完整事件，若果构成了一个完整事件则保存到数据库中，否则不保存，然后将该事件从事件集合中删除。如果在时间阈值范围内，那么需要判断该日志属于哪些事件，并更新相关事件，并记录事件类型。最后新建未被记录的事件类型加入到事件结合中，重新等待下一条日志。由于这两种安全事件计算方式相同，在代码实现上几乎完全相同，于是同样使用Java内置的反射机制来实现，以便于消除重复代码。

对于敏感端口，实则是一种扫描的流程，先将sensitiveport表中的配置的所有敏感端口信息读出到内存，并针对每一条数据进行匹配，一旦匹配到，则写入数据库。

4.3.4 日志关联分析

Apriori算法作为一种经典的关联规则挖掘算法[9]，在日志分析领域有着广泛的应用。在第二章第三节中介绍了Apriori算法。由于Apriori算法需要多次扫描数据库，运算复杂度较大。因此使用改进的Apriori算法进行防火墙日志之间的关联度分析，找到频繁项集[10]，挖掘潜在的连续。由于防火墙日志的规模十分庞大，所以不能通过人工分析，而且关联规则算法能够在可接受的时间范围内分析得出频繁项集。然后将支持度和置信度合格的结果，通过Hibernate提供的方法保存到数据库中，供给展示端展示。

由于防火墙日志是线性的并不是类同于商品的购买有天然的个体之间的集合关系。因此需要选定划分策略，最直观的划分策略是使用时间顺序对防火墙日志进行划分。在实际中，我们选定1秒钟内的所有的防火墙日志同属于一个集合。在进行关联度分析时，主要针对目的IP进行关联分析，并且对于目的IP相同的日志出现在同一集合中的情况，我们只需要保存一条即可。该系统的关联分析计算主要针对目的IP，因此对于一条日志，只选取其目的IP字段和时间戳字段，便于节省内存空间。将所有集合中的所有目的IP归入同一个目的IP集合。该目的IP集合包含了关联分析计算选取时间间隔内的所有目的IP。

同时设定支持度，只有当符合支持度要求时才是一个频繁结合。在实现Apriori时，针对不同的日志格式建立日志格式存在差异，因此为了后续的扩展性使用回调的形式增加可扩展性。

4.4 展示端实时分析

4.4.1 日志流量计算

在展示端需要获取1分钟的日志流量情况，由于展示端的ORM框架采用了Mybatis,在数据库的查询方式上，同分析端有一定的区别。我们使用Mapper注解进行数据库映射。Mybatis支持原生的SQL语句，便于SQL的编写和调试。

在展现方式上，使用动态折线图，能够生动直观显示日志流量的变化。因此使用C3.js提供的折线图。在使用C3.js之前需要先引入相关的依赖，分别是<script src="assets/js/lib/d3.min.js"></script>

<script src="assets/js/lib/c3.min.js"></script>。通过JQuery提供的Ajax方法从后台获取到实时的日志流量数据，并绘制到前端的折线图上。为了实现动态折现图的效果，需要使用javascript提供的Setinterval方法，来定时获取并且重新绘制折线图，于是在直观感受上，实现了动态折线图的效果。

由于需要定时重新绘制折线图，这其实是一个十分耗费资源的行为，于是借鉴ReactJs的思想，先判断重后端获取到的数据与上次绘制折线图的数据是否相同，如果相同，则不重复绘制相同的折线图，否则，才重新绘制折线图。经过这样的处理，能够大大减低浏览器的CPU使用情况。最终的实现效果如图4-2所示。

图4-2日志流量动态折线图

4.4.2 日志多种统计量统计

在前端展现统计结果时，采用了C3.js提供的表格控件，该控件与HTML自带的表格相比，更加美观，并且可以自定义类属性，实现表

格的不同风格。在绘制表格时，使用AJAX方法从后端获取相关数据，并绘制在前端页面上。其效果如图4-3所示。

图4-3 统计结果

4.5 展示端访问控制

在实现访问控制之前，需要实现用户注册和登录功能。登录和注册的实现效果如图4-4和图4-5。

图4-4 注册页面

图4-5 登录页面

由于本系统展示端提供RESTful API，是无状态的，是直接暴露给用户的。但作为一个完整的系统有必要进行必要的访问控制。本系统采用JSON Web Token进行授权和认证。在用户登录时，后台系统计算出一个用户token，而后访问相关RESTful API时需要携带该token。图4-6是用户请求RESTful API时携带的认证信息。

图4-6 携带token的请求头

4.6 展示端安全事件判定

在前端展现统计结果时，同样采用了C3.js提供的表格控件。在绘制表格时，使用AJAX方法从后端获取相关数据，并绘制在前端页面上。根据时间开始时间的倒叙进行分页。表格效果如图4-7事件统计前端实现图所示。

图4-7事件统计前端

点击查看详情按钮，可以显示该事件由哪些日志组成。事件详情图如图4-8所示。

图4-8 事件详情

由于事件一页难以显示完整，于是实现了分页效果。分页在后端采用SQL语句的形式实现。SQL语句如下：Select("select from event ORDER BY startTime DESC LIMIT {fromId}, {itemNum}; ")。在第一页和最后一页进行判断。其效果如图4-9所示。

图4-9 分页效果

4.7 展示端关联分析

关联分析展示端的实现和安全事件判定的实现较为相似，都是采用表格的形式。关联分析结果展示如图4-10所示。

图4-10关联关系结果

4.8 展示端日志记录

日志记录是记录当时发生的安全事件的一个重要手段，因此提供日志记录，查询，修改的功能是十分重要的，日志记录不涉及到分析端，只是展示端的实现。。因此采用现有的插件是经济且快速的方式。常用的富文本插件有UEditor、kindeditor、simditor和bootstrap-wysiwyg等。由于前端框架由bootstrap搭建于是选用bootstrap-wysiwyg插件来实现富文本效果。最终通过分页列表的形式展示日志。分页列表效果如图4-11所示。

图4-11日志记录列表

点检查看详情按钮可以查看日志详情并修改。其效果如图4-14所示。

图4-12日志详情

新建日志和日志详情前端实现相似，区别在于后端的实现，主要在于数据库的操作，新建时需要插入新的记录，而日志详情修改时只需要更新相关日志。

4.8系统测试

软件测试是软件系统开发流程中十分重要的一环。完整良好的软件测试能够提高软件运行的稳定性，提高软件的质量，并同时影响着用户的使用体验。

4.8.1 单元测试结果

由于分析端和展示端都使用Java语言开发，在分析端由于不需要使用Web功能，因此使用原生的Junit进行单元测试。在展示端需要使用Web功能，使用Junit和Spring boot Test模块进行单元测试。

测试结果如表4-3所示。

表4-3单元测试结果

子系统 使用测试模块 结果

分析端展示端 Junit4Junit4+Spring boot test 所有单元都通过测试所有单元都通过测试

4.8.2 集成测试

集成测试分为功能测试和性能测试。功能测试结果如表4-4所示。

表4-4 集成测试功能测试结果

子系统功能点 测试结果

分析端-实时分析分析端-安全事件判定分析端-关联分析展示端-实时分析展示端-安全事件判定展示端-关联分析展示端-日志记录 能够统计日志流量，统计多个字段发现多种网络安全事件计算出源IP之间的关联关系展示分析结果展示安全事件展示关联分析结果安全日志功能

性能测试结果如表4-5所示。

表4-5 集成测试性能测试结果

子系统功能点 测试结果

分析端-实时分析分析端-安全事件判定分析端-关联分析展示端-实时分析展示端-安全事件判定展示端-关联分析展示端-日志记录 1分钟处理200条记录1分钟处理200条记录

历史分析无性能要求渲染时间1秒钟以内符合实时渲染要求符合实时渲染要求

符合实时渲染要求

4.9本章小结

本章针对防火墙日志分析系统的分析端和展示端，在架构选择和实现细节进行了详细描述。并将实现结果展示出来。在完成系统后进行了必要的系统测试，以保证系统的稳定性。

参考文献

[1] Sarin, Deore. Survey Denial of Service classification and attack with Protect Mechanism for TCP SYN Flooding Attacks[J]. International Research Journal of Engineering and Technology (IRJET), 2016, 03(05): 1736-1738.

[2] 张大方,张洁坤,黄昆.一种基于智能有限自动机的正则表达式匹配算法[J].电子学报,2012,40(08):1617-1623.

[3] 张华强.关系型数据库与NoSQL数据库[J].电脑知识与技术,2011,7(20):4802-4804.

[4] Qrugantis S, Ding Q, Tabrizi N. Exploring Hadoop as a platform for distributed association rule mining[C] //The Fifth International Conference on Future Computational Technologies and Applications. Valencia: IEEE, 2013:62-67

[5]刘丽娟.改进的Apriori算法的研究及应用[J].计算机工程与设计,2017,38(12):3324-3328.

[6] 毛颖玲.利用动态创建script技术解决静态网站跨域问题[J].邵阳学院学报(自然科学版),2012,9(04):29-32.

[7] 杨传栋,曲洋.JSONP漏洞自动挖掘方法研究[J].电脑知识与技术,2016,12(35):17-18+21.

[8] 丁春玲,路志强,彭伟.Java反射机制在数据持久层轻量级ORM框架中的应用研究[J].西安文理学院学报(自然科学版),2017,20(01):39-42.

[9]赵洪英,蔡乐才,李先杰.关联规则挖掘的Apriori算法综述[J].四川理工学院学报(自然科学版),2011,24(01):66-70.

[10]Youcef Djenouri,Marco Comuzzi. Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem[J]. Information Sciences,2017,420.

[11]郑斌,沈明霞.在线富文本公式编辑器的设计与实现[J].计算机工程,2011,37(18):287-289.

[12]徐开勇,龚雪容,成茂才.基于改进Apriori算法的审计日志关联规则挖掘[J].计算机应用,2016,36(07):1847-1851.

[13]何月顺. 关联规则挖掘技术的研究及应用[D].南京航空航天大学,2010.

[14]李玉峰,杨婷,卜永波.Linux下基于Netfilter/Iptables防火墙的研究与应用[J].内蒙古农业大学学报(自然科学

版),2012,33(01):198-200.

[15]申彦. 大规模数据集高效数据挖掘算法研究[D].江苏大学,2013.

致 谢

论文写着也就快写完了，在完成毕业设计的过程中，经历了很多，要感谢很多人。尤其是我的指导老师刘雪娇副教授。刘老师以其过硬的专业知识、严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，严于律己、宽以待人的崇高风范，朴实无法、平易近人的人格魅力对我的影响深远。刘老师的悉心指导和帮助使我能够顺利完成毕业设计和论文。能够在刘老师的实验室经历一年时间的学习，不仅仅提高我的专业能力，包括开发能力和文档能力，更使得我能够在遇到困难时坚持下来并最终完成了本次毕业设计的工作。

同时我也要感谢我的同学，在我遇到困难时，他们给了我很大的支持，使我能够最终走完大学四年的学习生涯。匆匆四年，我经历了很多，从大一时对于计算机专业的一无所知，到现在学完了所有的本科课程，并有希望进一步深入学习，这都离不开老师和同学们的帮助。

感谢我的家人对我学习的支持，家人总是在我身后默默付出，帮助我克服各种困难。

最后，感谢论文评审老师的辛勤付出。

PAGE