编号：



**毕业设计说明书**

题 目： **基于Nginx的WAF的设计**

**与实现**

学 院： 计算机与信息安全学院

专 业： 信息对抗技术

学生姓名： 梁 青 浩

学 号： 1900310117

指导教师： 陶晓玲

职 称： 教授

**题目类型：**  理论研究 实验研究 工程设计 工程技术研究 √软件开发

√

2023 年 5 月 8 日

摘 要

随着互联网技术的普及和个人移动设备的保有量的提高，Web应用已成为人们生活中不可或缺的一部分。然而，随着Web应用的使用量增加，安全性也面临着越来越大的威胁，尤其是各种Web攻击威胁的出现，这些攻击可能会导致Web应用的服务中断、用户数据泄露和其他安全问题。为了保障Web应用的安全性和稳定性，本课题设计并实现了一种基于Nginx的Web应用防火墙。该防火墙可以有效地防止各种Web攻击，并提高Web应用的安全性和稳定性。

WAF是一种专业的Web安全工具，能够过滤、监控和阻止Web应用之间的HTTP通信流量，防范多种安全威胁，例如SQL注入、跨站脚本、XML外部实体、跨站请求伪造、Cookie篡改和CC攻击等。与传统防火墙和入侵检测系统相比，WAF的安全保护能力更为强大，能够有效地解决Web安全问题，为Web应用的安全运行提供保障。因此，WAF具有重要的研究价值和实际应用价值。本课题将会围绕以下几个方面进行设计与实施，1、通过正则匹配的形式实现Web攻击检测系统，能够较为自由的定义检测规则，能够从 Web 请求中的 IP、URI、请求参数、请求头等角度自定义白名单规则和黑名单规则，实现高效有益的防护，2、实现了一种改进的加权最小连接法的负载均衡算法。通过连接数与权值共同决定负载均衡过程中服务器的选着，应对高并发场景是性能问题，3、设计并且实现WAF的管理系统，WAF管理员与使用者能够实时查看WAF的日志记录，修改攻击检测规则与黑白名单IP等功能。

本题的选着切合现阶段的互联网面临的压力，选题新颖，对本系统进行调试与分析发现，系统能基本满足Web的防护，效果较好，具有重要的研究意义和实用价值。

**关键词**：Web安全；Web应用防火墙；Web攻击；Nginx；负载均衡

**Abstract**

With the popularization of Internet technology and the increasing ownership of personal mobile devices, Web applications have become an essential part of people's lives. However, as the usage of Web applications increases, their security is facing greater threats, especially with the emergence of various Web attack threats. These attacks may cause service interruptions, user data leakage, and other security issues. In order to ensure the security and stability of Web applications, this project has designed and implemented a Web Application Firewall (WAF) based on Nginx. This firewall can effectively prevent various Web attacks and improve the security and stability of Web applications.

WAF is a professional web security tool that can filter, monitor, and block HTTP communication traffic between web applications to prevent various security threats, such as SQL injection, cross-site scripting, XML external entities, cross-site request forgery, cookie tampering, and CC attacks. Compared with traditional firewalls and intrusion detection systems, WAF has a stronger security protection capability, which can effectively solve web security problems and provide security guarantees for the safe operation of web applications. Therefore, WAF has important research value and practical application value. This project will be designed and implemented around the following aspects: 1) implementing a web attack detection system in the form of regular matching, which can freely define detection rules and customize whitelist rules and blacklist rules from the perspectives of IP, URI, request parameters, request headers, and other aspects of web requests to achieve efficient and effective protection; 2) implementing an improved weighted least connections load balancing algorithm, which selects servers during the load balancing process based on both connection numbers and weights to address performance issues in high-concurrency scenarios; 3) designing and implementing a management system for WAF, which allows WAF administrators and users to view WAF log records in real-time, modify attack detection rules, blacklist/whitelist IP, and other functions.

The topic selection of this project is in line with the pressure faced by the current Internet, and the topic is innovative. Through debugging and analysis of the system, it is found that the system can basically meet the needs of Web protection and has good effectiveness, with important research significance and practical value.

**Keywords:** Web Security; Web Application Firewall; Web Attack; Nginx; Load Balancing.

目 录

[摘 要 2](#_Toc186)

[Abstract 3](#_Toc30174)

[第一章 绪论 6](#_Toc9961)

[1.1 研究背景及意义 6](#_Toc1560)

[1.2 研究现状 7](#_Toc26248)

[1.3 研究内容 8](#_Toc3180)

[1.4 论文组织 8](#_Toc16787)

[第二章 WAF相关技术分析 9](#_Toc30805)

[2.1 WAF相关技术分析 9](#_Toc12635)

[2.1.1 WAF 工作原理 9](#_Toc3737)

[2.1.2 WAF 部署方式 10](#_Toc15707)

[2.2 Web攻击威胁以及防御措施分析 11](#_Toc32591)

[2.2.1 Web攻击方式 11](#_Toc10677)

[2.2.2 注入攻击 12](#_Toc2286)

[2.2.3 失效的身份认证 13](#_Toc628)

[2.2.4 XSS跨站脚本攻击 13](#_Toc30971)

[2.3 Web攻击检测技术分析 14](#_Toc12813)

[2.3.1基于访问数据分析 14](#_Toc7957)

[2.3.2基于访问行为分析 14](#_Toc1203)

[2.4 Nginx服务器研究 14](#_Toc11628)

[2.4.1 Nginx架构与运行过程分析 15](#_Toc18293)

[2.4.2 Openresty的研究与分析 16](#_Toc6634)

[第三章 需求分析 17](#_Toc26957)

[3.1 用户需求 17](#_Toc28047)

[3.2 功能需求 17](#_Toc30074)

[第四章 Web应用防火墙的设计 19](#_Toc7124)

[4.1 系统整体结构设计 19](#_Toc14817)

[4.2 数据库选择分析 20](#_Toc2584)

[4.2.1 主流数据库分析 20](#_Toc16716)

[4.3.2 Mongodb服务器的优势 20](#_Toc28502)

[4.3 WAF模块设计 21](#_Toc18617)

[4.3.1 正则检测模块与响应模块 21](#_Toc14012)

[4.3.2 日志处理分析模块 22](#_Toc20511)

[4.3.4 负载均衡模块设计 22](#_Toc9185)

[4.4 WAF管理系统设计 23](#_Toc10829)

[4.4.1 数据展示模块 23](#_Toc11681)

[4.4.2 日志管理模块 24](#_Toc19664)

[4.4.3 用户管理与权限控制模块 24](#_Toc27885)

[4.4.4 WAF管理模块 24](#_Toc26632)

[4.4.5.后端功能设计 25](#_Toc10003)

[第五章 Web应用防火墙的实现 26](#_Toc20684)

[5.1 系统环境 26](#_Toc30536)

[5.2 数据库实现 29](#_Toc20559)

[5.2.1 数据库搭建 29](#_Toc31096)

[5.2.2 数据库集合与数据实现 30](#_Toc21550)

[5.3 规则匹配模块实现 31](#_Toc10413)

[5.5 日志处理分析模块实现 34](#_Toc1984)

[5.6 负载均衡模块实现 34](#_Toc30086)

[5.7登录与前后端交互 35](#_Toc26919)

[5.7.1登录流程 35](#_Toc32152)

[5.7.2 前后端交互实现 36](#_Toc16311)

[5.8 数据展示页面实现 37](#_Toc21171)

[5.9 日志管理模块实现 38](#_Toc18083)

[5.10 权限控制模块实现 38](#_Toc31065)

[5.11 WAF管理模块 39](#_Toc13847)

[第六章 Web应用防火墙测试与验证 41](#_Toc16365)

[6.1可行性分析 41](#_Toc26153)

1. 绪论
   1. 研究背景及意义

在当下互联网影响下，Web服务在生活中有着十分广泛的应用，影响着生活的各个方面与各个行业，有着繁多的应用场景，无论是电子商务、社交生活、技术论坛，个人博客、政企工作、游戏娱乐、电子邮件管理、系统后台管理等等。而每天有着巨大的互联网访问量，根据中国互联网络信息中心（CNNIC）在北京发布的第51次《中国互联网络发展状况统计报告》显示，截至2022年12月，我国网民规模已达10.67亿，与去年2021年的12月相比人数增长达3549万人次，互联网普及率达到了惊人的75.6%。

随着新型基础设施建设持续推进，我国网络基础能力不断增强，万物互联基础不断夯实，工业互联网体系构建逐步完善，互联网应用用户规模保持稳定，这个数字也将会不断地提升。这数字的背后不仅仅意味着欣欣向荣的社会发展，还隐藏着一些列安全隐患，正因为Web技术凭借其标准化与通用化的特点取得广泛应用，越来越多人通过互联网访问Web服务，但在飞速发展的同时，其本身的安全形势并不让人乐观，据新华三《2021年网络安全漏洞态势报告》显示，将新增漏洞规则按照攻击对象进行统计，Web应用类漏洞占比较高，达到48.1%，而Acunetix公司 2019 年的 Web 应用漏洞报道显示，多达 46%的网站存在高等严重程度的漏洞和风险，有 87%的网站存在中等严重程度的漏洞。而一旦Web服务器因为攻击而停止服务，带来的经济损失是巨大的，并且对于使用者的数据泄露影响也将是致命的。

对于这种出现在OSI（Open System Interconnection）中应用层的安全隐患，传统计算机网络防火墙能发生的作用就十分有限了，因为传统的防火墙工作场景主要是网络层次，主要原理是针对IP报文进行检测与分析，一般情况下只对端口进行限制，并且在普通防火墙产品在设计之初就不需要理解HTTP（S）会话，这也就决定了无法理解Web应用程序语言的事实，而80（443）端口作为Web应用发布端口，基于其必须对外开放的原则，从而导致Web攻击行为可直接绕过传统防火墙的检测。因此针对传统防火墙功能缺陷的问题，就提出了新型的Web应用防火墙（Web Application Firewall，WAF）。专门用于解决HTTP应用端口的安全问题。

* 1. 研究现状

2004年起WAF的概念就已经被国外的安全厂商提出，至此开始初步尝试，例如梭子鱼网络有限公司将Netcontinuum公司纳入旗下，当时的Netcontinuum就是这一领域的先行者，其解决方案包含网站的网络应用安全、通信管理和SSL加速等，支付卡行业安全标准委员会也发布了支付卡行业数据安全标准（PCI DSS），这让Web应用防火墙开始被越来越多的人所熟知。但因为业界一直缺乏明确的标准，有些安全产品也能够在一定层次上保护某些Web应用，因此人们对于其概念比较模糊，而随着互联网行业的兴起，Web服务广泛出现在各个行业当中，而信息安全问题逐渐凸显，WAF的概念开始走进大众视野

对于WAF的研究，国外研究WAF的历史较长,研究成果更加丰富。包括WAF的分类、检测技术、防护策略等方面的研究都更加全面和深入。国内相关研究起步较晚,成果还相对较少。国外研究更加重视实证与数据分析,提出的方法或策略常常有详尽的实验来验证有效性。国内相关研究在这方面还较为欠缺,较多停留在理论阶段。

WangQian等人根据OWASP（开放Web应用安全）发布的数据，分析得出访问控制漏洞已成为十大Web应用安全漏洞之一，提出了一种基于上下文的检测方法来检测访问控制漏洞,而不是依赖于访问控制策略规范。该方法使用授权上下文的概念来描述每个安全敏感操作,并通过比较同一操作的授权上下文的不一致性来发现访问控制漏洞，该方法避免了提取访问控制策略这一难题,而是通过比较授权上下文来发现漏洞。该方法在一定程度上推进了网页应用访问控制领域的研究。

Prateek Gupta, Piyush Mittal等人提供了网络应用防火墙(WAF)的概况，介绍了WAF的分类，WAF的工作机制， WAF产品比较。比较了几种主流WAF产品,如ModSecurity、DotDefender、Cisco ACS等。这些产品有不同的部署位置、支持的web服务器以及定价模式。 WAF的优点与局限性。WAF的主要优点是增强web应用程序安全、降低维护成本。主要局限性是效率降低、误报率高和无法检测0day漏洞等 WAF的发展趋势，未来WAF可能采用机器学习检测web攻击、提供更丰富的检测规则、支持更多 web 应用框架和添加云WAF等比较全面地概述了WAF技术。  
 Hyunki Kim与其团队提出一种基于树的状态机模型来表示SQL查询，基于构建的状态机模型,设计了一种SQL注入检测方法，对其进行了评估。评估结果表明,该方法可以检测典型SQL注入攻击,如联合查询、子查询等,其检测准确率达到95%。与其他方法相比,该方法具有较高的检测覆盖率和较低的误报率，不足是它目前仅支持MySQL,对其他数据库的支持还需要进一步研究。

就以上的国内外学者的研究不难看出，对于WAF的研究与发展中，学者大多把目光放到异常的访问与攻击检测方式上，有根据主流的攻击提出上下文检测方法，有根据市场的需求提出不同云WAF、还有根据传统的SQL注入攻击设计检测模型等等，但这些研究较少把目光放到WAF整体性能问题上，对于日志分析与数据展现缺少关注，然而随着Web安全观念的普及，普通使用者也需要看到鲜明的拦截效果打消个人的顾虑，因此研究一个功能完善，数据展示清晰明了，易于使用者查看的WAF及其管理系统是及其必要的。

* 1. 研究内容

如今商用WAF不在少数，其功能强大，能抵御各种攻击，但是仍存在不足之处，例如，部分WAF服务器直接与Web服务器想来，部署于同一服务器中，这一行为较为考验web服务器性能，有较高的耦合性，在维护过程中存在风险，除此之外，绝大多数的WAF部署方式是单点部署，这将会导致如果WAF设备发生故障,那么web应用程序将完全暴露在攻击之下,安全性无法得到保证，导致响应时间增加和吞吐量下降，可扩展性下降等问题，因此就以上问题进行改进。

本课题主要的研究内容如下：

1. 根据已有WAF进行调研，吸收其优点，分析其不足，了解其工作原理，研究常见的Web攻击以及响应的防御手段，Web攻击检测的方法与Nginx关键技术
2. 设计基于正则匹配规则与黑白名单防护的混合型攻击检测模块，该模块能自定义配置检测规则与黑白名单IP进行有效防护。
3. 研究并实现一种加权加权最小连接数的负载均衡算法，凭借各个服务器连接数与其权值作为参考指标，决定下回访问分配的服务器，从而均匀化各个服务器负载。
4. 设计并实现高度友好化的WAF管理系统，该系统在权限方面进行设计，保护自身安全的前提下，管理员与用户可以访问该系统，但是由于角色不同所能访问的页面也不同，能实现日志记录查看，了解访问数据与攻击数据，对WAF的匹配规则集以及黑白名单进行配置。
   1. 论文组织
5. 绪论。本章首先分析了 WAF 的研究背景及意义，接着介绍WAF的国内外发展现状，主要研究方向的突破与关注度不高但较为重要的方面，并根据此提出本课题的优势，之后明确了本次课题的研究内容与方向。
6. WAF相关技术分析，本章介绍完成整个课题需要了解的关键技术，从WAF的工作原理与部署原理出发，随后分析当前互联网主要面临的攻击威胁以及对应的防御手段，随后针对异常检测进行研究用于攻击检测模块，最后分析部署WAF用到的Nginx的核心科技以及在其基础上实现lua模块化开发的OpenResty的分析。
7. 需求分析及设计思路，本章从用户面临的痛点进行分许，总结普遍用户需要，根据用户的需要得出功能需求，按照既定的功能进行得出设计思路随后进行较为合理的研究与设计。
8. Web应用防火墙的设计，本章主要阐述本课题WAF系统的整体设计方案，随后对数据库的选着进行论述，再对WAF的正则检测模块，功能拦截模块、日志处理分析模块、载均衡算法进行功能分析与结构设计，随后根据用户需求对WAF管理系统的各个模块进行设计。
9. Web应用防火墙的实现，本章就系统环境、数据库搭建、WAF模块的实现、WAF管理系统的实现等进行论述，其中附带一些核心代码，分析代码的逻辑与理论功能。
10. Web应用防火墙测试与验证，本章就完成的系统进行功能评测与性能评测，以数据论证课题的完整性与实用性，而测试结果表明实现的系统基本符合课题的要求，有较强的实用性。
11. 结论，本章总结本课题主要工作，分析存在的不足，并且对后续研究进行一个方向思考。
12. WAF相关技术分析

本章介绍完成整个课题需要了解的关键技术，从WAF的工作原理与部署原理出发，随后分析当前互联网主要面临的攻击威胁以及对应的防御手段，随后针对异常检测进行研究用于攻击检测模块，最后分析部署WAF用到的Nginx的核心科技以及在其基础上实现lua模块化开发的OpenResty的分析。

2.1 WAF相关技术分析

2.1.1 WAF 工作原理

WAF的运行机理是在根据事先定义的安全防护规则集与黑白名单进行防护，对所有向受保护的Web服务进行访问请求和响应过程进行一个规则匹配以及内容隔离。以便保护Web服务器，与传统防火墙对比，WAF是专门防护web应用程序有着更高的专业性。传统防火墙作为网络中最常用的安全设备之一，可谓是无处不在，其工作部署在OSI七层模型当中的网络层与传输层，而关于高于网络层的攻击安全策略就略显不足，特别是如今绝大多数网络攻击已经与传统的一些针对本地pc机的入侵不同。更多的是发生在应用层上，以各种HTTP流量的方式，这将会涉及传统防火墙的防御盲区。

传统防火墙的主要防护手段有包过滤、状体监测、代理服务提供的安全防御机制是一种被动的安全模式。传统防火墙采用逐一匹配的方法, 使得计算量特别庞大。安全性与检查的成正比, 高安全性意味着低效率, 最终导致网络传输速率减慢, 成为防火墙内外网络传输的瓶颈。安全性与效率成反比。[罗广华浅谈WAF]

对比传统防火墙存在的问题，WAF有较好的处理，首先本作作为专业Web服务安全防护工具，WAF工作在应用层，能够针对HTTP请求以及响应做出较为完整的分析与检测，通过重组、解析、来达到理解HTTP请求和应答的目的，包括对HTTP请求头、请求体、响应头与响应体等进行检查。根据前置的Web攻击特征识别达到基于HTTP网络攻击的监测与阻隔。在监测到异常后能够马上作出反应，如报警、阻断、记录等。而正是通过前置的一系列规则，WAF能够实现规则匹配达到过滤恶意流量防止程序漏洞，使得Web服务器免于收到SQL注入攻击、跨站脚本（XSS）、跨站伪造（CSRF）攻击，假如加上同一IP访问密集度的统计，还能实现防御CC攻击。

通过以上论证，我们不难看出，WAF的核心架构可以分为入侵检测模块、响应模块。入侵检测模块通过检测访问流量的携带信息与前置的规则进行匹配拦截，而前置匹配模块可以由攻击匹配规则和黑白名单集共同组成，提高WAF的工作效率与安全稳定性，而响应模块就是当检测到攻击时的处理。当然除此之外WAF还能添加许多模块，如管理模块、日志模块、负载均衡模块等，本文在保证核心功能的基础上，将会考虑日志模块、管理模块与负载模块的设计。

2.1.2 WAF 部署方式

WAF的部署一般支持透明代理，反向代理，旁路监控，桥模式部署。透明代理部署模式支持透明串接部署方式。串接在用户网络中，可实现即插即用，无需用户更改网络设备与服务器配置。部署简单易用，应用于大部分用户网络中，

采用旁路监听模式，在交换机做服务器端口镜像，将流量复制一份到 WAF 上，部署时不影响在线业务。在旁路模式下 WAF 只会进行告警而不阻断。透明桥模式是真正意义上的纯透明，不会改变更改数据包任何内容，比如源端口、TCP 序列号，桥模式不跟踪 TCP 会话，可支持路由不对称环境。WAF采用反向代理模式以旁路的方式接入到网络环境中，需要更改网络防火墙的目的映射表，网络防火墙映射WAF的业务口地址，将服务器的IP地址进行隐藏。如下图2-2.1所示

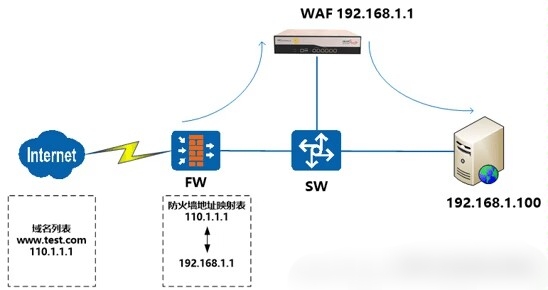
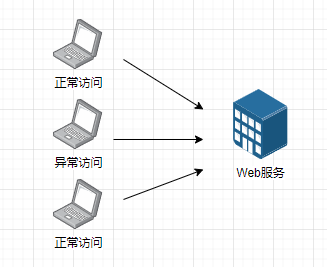
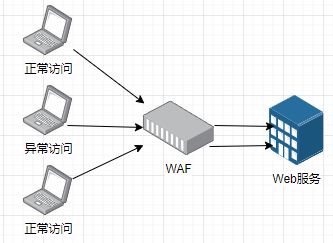


图2-1.1反向代理部署图

而以上四种方式中反向代理的方式是最安全的，反向代理使得访问流量无法直接知道Web服务器的真实ip，只能通过代理服务器代理转发请求。而且反向代理支持负载均衡，能够支撑高并发问题，保证访问效率。考虑到WAF的各方面性能，本文决定使用反向代理方式代理WAF。

有无WAF的Web服务示意图如下图2-1.2图所示 图（a)表示典型的Web服务架构，在典型架构中，请求能够直接到达服务，无任何防护措施，黑客能直接进行攻击，这无疑是不合理。图（b）是通过反向代理部署WAF的Web服务结构，在当前结构下，所有访问必须经过WAF，在WAF中进行流量分析与审计、只有通过了审核的流量才能经过WAF转发到Web服务，反之将被直接阻隔。而分析的过程无疑会增加一些响应时间，所有如何提高审核速度也是设计当中需要考虑的一个关键。

1. 无WAF保护的Web服务架构 （b）有WAF保护的Web服务架构

图2-1.2WAF部署原来图

2.2 Web攻击威胁以及防御措施分析

2.2.1 Web攻击方式

开放式Web应用程序安全项目（OWASP，Open Web Application Security Project）是一个组织,在Web安全发展中起着重要作用，在网络安全的实践者当中有着良好的口碑。该组织每年都会整理当年在网络应用层中出现的漏洞，通过分析各个漏洞的危害等级，从2003年该组织首次出版了“Top 10”，也就是10项最严重的Web应用程序安全风险列表。Top 10 总结了Web应用程序最可能、最常见、最危险的十大安全漏洞，TOP 10的提出使得WAF的安全防范策略得到更合理的安全指导，对于WAF的发展是影响深远的。

通过分析不同年份威胁的排行我们可以更好的实现WAF的设计，下表2.1将2017年版本与2022年版本的OWASP Top10进行简单对比

表 2.1 OWASP Top 10 – 2017 与 OWASP Top 10 – 2022 对比

|  |  |
| --- | --- |
| OWASP Top 10 - 2017 | OWASP Top 10 - 2022 |
| A1:2017 – 注入  A2:2017 – 失效的身份认证  A3:2017 – 敏感数据泄露  A4:2017 – XML 外部实体（XXE）  A5:2017 – 失效的访问控制  A6:2017 – 安全配置错误  A7:2017 – 跨站脚本（XSS）  A8:2017 – 不安全的反序列化  A9:2017 – 使用含有已知漏洞的组件  A10:2017 – 不足的日志记录和监控 | A1:2022 – 注入  A2:2022 – 失效的身份认证  A3:2022 – XSS（跨站脚本攻击）  A4:2022 – 不安全的对象直接引用  A5:2022 – 跨站请求伪造（CSRF）  A6:2022 – 安全配置错误  A7:2022 – 限制URL访问失败  A8:2022 – 未验证的重定向和转发  A9:2022 – 应用已知漏洞的组件  A10:2022 – 敏感信息暴露 |

从表2.1中不难看出，注入攻击、失效身份认证、跨站脚本攻击一直都是比较普遍的攻击方式，除此之外，一些旧的攻击方式随着WAF的安全防护提升已经较少出现如，不足的日志记录和监控，当然也新增了一些新的攻击方式值得我们注意与研究、如限制URL访问失败，就上面常见的攻击我们在下文进行介绍与分析

2.2.2 注入攻击

注入攻击：注入攻击作为常年排在第一的攻击方式，给Web服务的安全带来十分巨大的影响，而注入攻击的基本原理是向应用程序输入恶意代码以非法访问或破坏系统，由于发送地内容会触发编译器执行未被授权地指令，读取特权数据或者破坏特权数据，导致系统破坏与数据泄露。常见的注入攻击类型有:SQL注入，LDAP注入，OS命令注入。

SQL注入:向SQL查询语句插入恶意SQL代码,以访问或破坏数据库。web应用程序通常使用动态SQL语句与数据库交互,即SQL语句中的某些内容来自用户输入。例如:

|  |
| --- |
| （String sql = "SELECT \* FROM users WHERE name='" + userName + "'";） |

这里,userName来自用户输入,如果用户输入包含 SQL 语法,则会破坏原SQL语句的语法,构成SQL注入攻击

LDAP注入:向LDAP请求插入恶意LDAP代码,以访问或破坏目录服务。web应用程序会使用LDAP协议与目录服务器交互,以验证用户身份或查询用户信息。这通常使用动态构建的LDAP请求,例如:

|  |
| --- |
| （String ldapURL = "ldap://server/ou=people,dc=example,dc=com?uid=" + inputUsername; ） |

这里,inputUsername 来自用户输入,如果包含 LDAP 语法,会破坏原LDAP请求,导致LDAP注入。

OS命令注入:向操作系统命令行插入恶意系统命令,以控制操作系统。web应用程序调用某脚本,脚本中包含来自用户输入的参数用于指定脚本行为。如果用户输入包含命令语法,则会破坏原命令或脚本语义,导致执行恶意系统命令。

例如,有一个网站验证码功能调用系统命令:

|  |
| --- |
| cmd = "convert captcha.jpg -fill '" + inputColor + "' -font '" + inputFont + "' -pointsize 30 -annotate +20+50 '" + inputText + "' captcha\_out.jpg"  os.system(cmd) |

这里,inputColor、inputFont和inputText来自用户输入。如果输入包含如（`;ls /`）等恶意命令,则会执行未授权的系统命令。

这些攻击的危害包括信息泄露、网页挂马、删除数据以及完全控制系统等。

而防御方法最为有效的就是检测所有请求参数，使用WAF检测注入:采用WAF设备或者规则,通过正则表达式、语法分析等方法检测注入攻击。

2.2.3 失效的身份认证

失效的身份认证指认证系统或过程存在问题,无法有效验证用户身份或权限,导致安全隐患，主要造成原因如下，身份验证不严谨、密码安全性不高、会话管理不当、权限管理不当等。

如果攻击者获得了其他用户的账号密码或令牌,则可以以该用户的身份访问系统与资源。这可能导致隐私泄露、数据泄露或滥用资源等问题，假如获得的是管理员用户对系统的威胁是毁灭性的。这些问题的根源在于认证过程或系统本身存在一定的缺陷或未考虑相关的攻击手段。

针对这类漏洞，要预防这些问题,需要从密码强度、注册验证、会话管理、权限控制等多个方面进行完善,构建一套严密的认证机制。

2.2.4 XSS跨站脚本攻击

XSS(跨站脚本攻击)是一种代码注入攻击。它允许攻击者将恶意脚本代码注入到网页上,当用户浏览网页时,嵌入其中执行恶意脚本，而这一过程将会导致信息泄露、网页挂马、BOT网络攻击、勒索软件感染等严重影响与损失。

一种常见的XSS攻击过程如，攻击者找到一个存在XSS漏洞的网页,可以通过查看网页源代码或使用开发者工具获取源代码，攻击者根据查找到的注入点,构造可以执行恶意行为的脚本代码，攻击者构造好恶意代码后,将其注入到网页的注入点中，当受害用户浏览含有恶意代码的网页时,嵌入其中的脚本代码将自动执行。这将触发XSS攻击,导致信息泄露、网页挂马等危害影响，如让攻击者获取用户的数据，如Cookies、会话ID等，既可以使用用户身份登录网站,也可以获取到用户隐私信息。

针对该漏洞，Web应用需要输入进行严格过滤，过滤<、>、%、'、"等特殊字符,或实体转换这些字符，过滤事件处理程序如onclick、onload等，过滤javascript:、vbscript:等代码执行协议， 对输出进行编码输出内容使用 HTMLEncode 进行编码,将特殊字符转换为HTML实体，使用HTTP Only Cookie，即（禁用JavaScript读取Cookie权限,防止XSS窃取用户Cookie），避免DOM XSS等方式进行。

2.3 Web攻击检测技术分析

针对Web攻击检测方法主要分为基于访问数据分析和基于访问行为的分析，通俗的介绍，基于访问数据分析可以有效的检测注入攻击，跨站脚本攻击，判断是否有异常请求数据，而基于访问行为分析，可以制约访问次数有效避免暴力破解以及CC攻击

[2.3.1基于访问数据分析](#_Toc32382)

基于访问数据分析主要需要把请求数据进行解析，获取请求头、请求体中的数据，随后先与事先定义的黑白名单进行对照，假如为黑名单IP直接交由响应系统进行处理，假如是白名单就让其通过，这样可以提高响应速度，得到一个较好的处理方式。假如访问ip并不在二者之中，那么就需要调用规则匹配模块，查看请求数据是否有非法字符与攻击嫌疑，假如有的话直接交由相应模块处理。

而正则匹配模块中的规则可以借鉴网上开源的权威的安全规则库 ModSecurity CRS 来扩充匹配规则，本课题为保证自定义规则的自由性，不直接在WAF中装载ModSecurity CRS规则集，而是让管理员借鉴其中规则进行规则集的填充。

[2.3.2基于访问行为分析](#_Toc26551)

基于访问行为分析主要通过统计访问的IP有一定时间内的重复访问次数的行为进行统计与做出响应，假如访问在一定时间内比较密集，则说明该IP存在爆破攻击与CC攻击的可能，因此需要通过响应模块对该IP进行一定时间内的封禁。

2.4 Nginx服务器研究

Nginx是基于C语言开发是一个高性能的HTTP和反向代理服务器,也是一个IMAP/POP3/SMTP代理服务器，自2004发布至今约莫20年，因为其高性能、高稳定和高扩展深受广大用户青睐。据 Netcraft 统计，截至2022年11月，

Nginx的服务器市场占有率达26.51%，依旧稳居榜首。

正因为采用事件驱动模型:采用epoll等高效事件模型,单线程就可以高效处理成千上万个并发连接。无阻塞IO:采用非阻塞IO模式,不会频繁系统调用,大大减少上下文切换。其优点可谓是企业与用户选着的重要因素

[2.4.1 Nginx架构与运行过程分析](#_Toc32382)

Nginx整体架构如图2-3所示

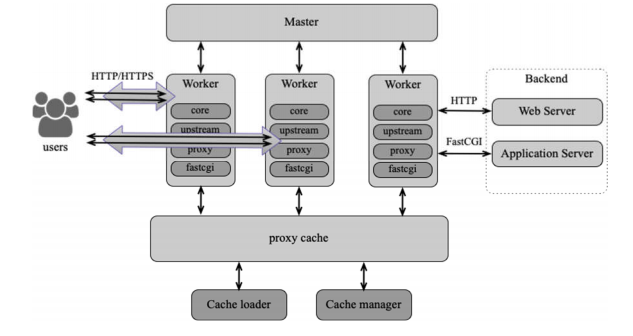


图 2-3 Nginx 整体架构图

如上图，在Nginx中Master进程可以称之为监控进程，总管多个Worker进程，而Worker进程的数量一般有CPU的内核决定，因此不同的核心数就决定了服务器的性能，Worker进程的工作为处理HTTP请求与响应的进程，以及同上游服务器通信。Cache Loader进程的主要作用是提前加载缓存内容,而通过Cache Manager进程进行定期管理，如此它的存在可以明显改善Nginx缓存服务器在启动初期的访问体验。而nginx采用MasterWorker的形式能够拥有低内存消耗，高cpu使用率的效果

其次Nginx支持及其自由的模块化，而自定义模块进行Nginx服务器配置也是备受广大使用者称赞的。这也是本课题使用Nginx服务器作为基础的原因。

而关注Nginx的运行过程是进行WAF开发尤为重要的一个过程，研究Nginx运行的各个阶段以及该阶段进行的工作，对于WAF的部署与改进算法的部署尤为重要。Nginx处理阶段如下图2.4所示。

图2.4 Nginx 处理 HTTP 请求的阶段

 Nginx 不仅实现了 HTTP 服务器的功能，并且还内置多种优秀的负载均衡器的功能如轮询法、加权、最小连接法、随机法、IP哈希法等等。可是上诉算法都是在静态下的，而在动态环境下，访问较为复杂，静态的算法无法发挥上游服务器最大性能，因此本课题提出改进的负载均衡算法，提高访问效率与系统性能，

2.4.2 Openresty的研究与分析

OpenResty 是一个基于 Nginx 与 Lua 的高性能 Web 平台,其内部集成了大量精良的 Lua 库、第三方模块以及大多数依赖项。它可以帮助我们构建大规模的 Web 应用、Web 服务和动态网关，优点较为明显如性能高，因为基于Nignx与LuaJIT,性能较传统Nginx+PHP等方案提高5-10倍，其次在openResty中可以自由配置Nginx与编写Lua脚本,实现各种Web服务器功能，在与数据库连接方面也有着巨大遍历。

最为主要的因素是其主要功能与Nginx相同，并且其集合了Lua模块因此在自定义模块的适合十分容易进行编写与装载，这一优势对于WAF的设计与装载无疑是优势巨大的，因此本课题决定使用OpenResty作为主要的部署方式进行WAF的反向代理以及负载均衡等功能。

1. 需求分析

本章主要对整个用户与系统的功能需求进行分析，先通过调查用户需求，确定系统的设计需求，明确了整个系统的大需求后细分功能需求

[3.1 用户需求](#_Toc17930)

发展速度迅猛的Web应用程序对于安全的要求无时无刻不在提高，也挑战者安全领域研究人员的能力与见识，虽然WAF不断更新迭代，安全性能不断提升，但是攻击者的能力也在不断加强，因此现阶段的Web服务的最根本需求就是安全的保障，其次才是各种便利快捷的服务。以下我们从个人与企业用户的角度分析各自的需求。

假如是对于个人，自己制作的一个个人博客或者介绍网站。对于WAF的需求可能是防止黑客入侵网站。需要的WAF首先得部署便捷，功能齐全，能够防止常见的Web攻击，sql注入、XSS攻击，具有较高性能还有可扩展性，能够处理博客流量，并且安装的流量花费不能太高，因为个人网站在服务器的选着上，带宽与大型企业网站相比是远远不及的，高流量损耗意味着较慢的响应。但是对于负载均衡的要求不会太高，访问IP与访问量不会在同一时间段中非常高，即并发量不高。

但是假如对于企业用户来说的话，需要关注的方向会更多，首先在保证安全前提下，需要提高访问速度，与并发负载的稳定先，网站访问的用户量较大，并且在同一时间段的处理是必须关注的，除此之外，提供易于使用的管理界面，以便企业可以轻松地配置和监控WAF，本身作为企业用户，网站的维护者不仅仅是一个人，而是一个团队，而团队当中每个人由于权限不同对于WAF管理的权限也不同，首先会有一个整个系统的管理员，具有整个系统的所有权限，其次是网页管理员，能够对网页有规则新建和黑白名单修改的功能，除此之外还有一个普通用户，只能够查看WAF的防护状况，不能进行规则的配置与黑白名单的添加修改。

[3.2 功能需求](#_Toc28097)

从用户需求得出功能需求如下，把课题需要实现的任务分为两部分。WAF部分与WAF管理系统，而WAF作为整个课题的根本核心

首先，WAF需要支持多种Web应用程序，包括常见的CMS（内容管理系统）和自定义应用程序。这是因为不同的组织和企业使用的Web应用程序可能不同，因此WAF需要支持多种应用程序，以便能够满足不同的需求。

其次，WAF需要能够检测和阻止常见的Web攻击，如SQL注入、XSS、CSRF、文件包含、命令注入等。这些攻击是Web应用程序面临的最常见的威胁，因此WAF需要能够有效地检测和阻止这些攻击。

第三，WAF需要具有高度可配置性，以便管理员可以根据其特定的安全需求进行自定义设置。这是因为不同的组织和企业可能有不同的安全需求，因此WAF需要具有高度可配置性，以便管理员可以根据其特定的需求进行自定义设置。

第四，WAF需要能够提供实时日志记录，以便管理员可以及时采取行动。这是因为及时采取行动可以减少攻击对Web应用程序的影响，并帮助管理员更好地了解攻击的性质和来源。

第五，WAF需要具有高性能和可扩展性，以便能够处理高流量的Web应用程序。这是因为Web应用程序可能会面临高流量的情况，因此WAF需要具有高性能和可扩展性，以便能够处理这些高流量的情况。

第六，WAF需要能够与其他安全工具和系统集成，以提供全面的安全解决方案。这是因为WAF只是安全解决方案的一部分，因此需要与其他安全工具和系统集成，以提供全面的安全解决方案。

第七，WAF需要提供详细的文档和支持，以便管理员可以快速解决任何问题或疑虑。这是因为管理员可能会遇到问题或疑虑，因此需要提供详细的文档和支持，以便管理员可以快速解决这些问题或疑虑。

第八，WAF需要具有合理的价格和许可证模型，以便各种规模的组织都能够使用它。这是因为不同的组织和企业可能有不同的预算和规模，因此WAF需要具有合理的价格和许可证模型，以便各种规模的组织都能够使用它。

而另一部分的WAF管理系统需要以下功能

用户管理：WAF管理系统需要支持用户管理功能，包括用户登录、权限管理等。管理员可以创建多个用户，并为每个用户分配不同的权限，以便实现安全的管理。

WAF配置：WAF管理系统需要支持WAF配置功能，包括规则配置、白名单配置等。管理员可以根据其特定的安全需求进行自定义设置，并对WAF进行配置和管理。

报告生成：WAF管理系统需要支持报告生成功能，包括攻击报告、日志报告等。管理员可以根据需要生成各种类型的报

身份验证：WAF管理系统需要支持身份验证功能，以便防止未经授权的访问。管理员可以配置身份验证规则，并监控用户的访问情况。

安全审计：WAF管理系统需要支持安全审计功能，以便对WAF的运行情况进行审计和评估。管理员可以查看WAF的日志记录，并对WAF的运行情况进行分析和评估。

易于使用：WAF管理系统需要易于使用，以便管理员可以轻松地配置、监控和维护WAF。管理员可以使用直观的界面和工具，以便快速完成各种任务。

1. Web应用防火墙的设计

本章的主要内容是对WAF系统整体进行设计，WAF的核心内容分为WAF部分与WAF管理系统两部分，WAF中分为正则匹配模块、响应模块、日志处理模块、负载均衡算法。而WAF管理系统分为数据展示模块、日志管理模块、权限分配模块以及WAF规则管理模块组成，结构清晰，为系如的实现提供明确参考

4.1 系统整体结构设计

本课题针对WAF的设计，在核心分为WAF和WAF管理系统两个大块进行分类，而各个大块的内部又分为各自的小块，功能相互关联协调，实现一个稳定高效的WAF系统。总设计图如下图4-1所示

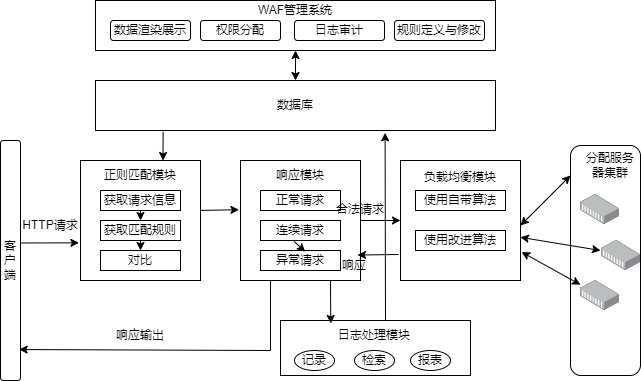


图4.1 WAF总设计图

首先在客户端发起HTTP请求后交由正则匹配模块进行处理。正则匹配模块先从请求中提取请求的特征参数例如请求体、请求头、客户机IP、请求方法等信息，随后在通过数据库获取黑白名单、匹配规则等，随后根据数据库返回的数据。对比请求内容，分辨是正常请求还是异常请求。假如请求是正常的就推给相应模块，响应模块还会再进行一次辨别，假如请求的IP的一定时间内多次访问，则以而已访问进行处理。假如正常请求则通过负载均衡模块进行处理，负载均衡模块再通过负载算法。将请求均衡的分发到各个上游的Web服务器节点，以及通过日志处理模块进行记录。相应的网站在做出响应，。而异常请求在被日志处理模块记录后将会直接跳转到指定的网站。

在整个流程中无论是何种访问都会通过被日志处理模块记录，使得WAF管理者能通过分析日志判断网站的安全，除此以外，在最上层的WAF管理系统与WAF的各个阶段也是密切相关的。在WAF管理系统中，管理员能够通过可视化的数据渲染，清晰的分析web的访问情况与安全状况，还能够通过自定义规则与自定义黑白名单对WAF进行管理。

4.2 数据库选择分析

在整个系统设计中，数据库尤为重要，数据库起到连接WAF管理系统与WAF的决定性作用。无论是日志存储还是规则存储都是核心点，可以说数据库假如出现问题那么整个系统将会陷入瘫痪。

[4.2.1 主流数据库分析](#_Toc18891)

当先的Web服务中最为重要的就是数据库了，互联网时代也可称之为数据时代，可见数据的重要性，而数据的载体数据库就更为重要了。主流的数据库系统有很多种。其中包括关系型数据库、非关系型数据库和分布式数据库。

首先，关系型数据库是最常见的数据库类型，它们使用结构化查询语言（SQL）进行数据操作。其中，最著名的关系型数据库有MySQL、Oracle和SQL Server。关系型数据库的优点是数据结构清晰，易于维护，适用于存储结构化数据。

Oracle数据库。这个强大的数据库系统以其稳定性和可扩展性而闻名，成为大型企业的首选。它具有诸如高级安全性和高可用性等出色功能，有助于保护数据安全。

MySQL。这个开源数据库非常受欢迎，尤其是在Web应用领域。它易于使用，拥有庞大的开发者社区，不断致力于改进。

SQL Server。这个数据库在依赖微软技术的企业中非常受欢迎。它具有许多有助于管理大量数据的功能，如数据仓库和商业智能工具。

其次还有非关系型数据库(NoSQL),其与传统的关系型数据库(SQL)相比，模式是灵活的、不需要预定义的模式。这适用于模式变更频繁的情况，存储结构简单,没有表与表之间的连接关系概念。这有利于非常大量数据的高效存储与查询，其中典型的如MongoDB，也江苏我们本课题所使用的数据库。

[4.3.2 Mongodb服务器的优势](#_Toc18891)

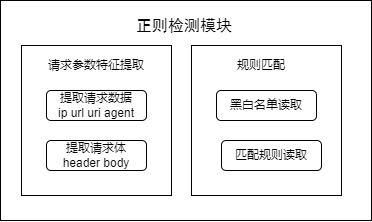
[我们来看看MongoDB。这个数据库与众不同，因为它是一个NoSQL数据库，意味着它专为处理非结构化数据而设计。它非常灵活，可以水平扩展，非常适合大数据和实时应用。](#_Toc18891)

[而对于管理系统而言MongoDB是一种无模式的文档型数据库,不需要预定义严格的表结构。这使其非常适合页面的的数据渲染,即使数据结构频繁变更也能轻松应对。并且在返回的数据类型上，Mongodb支持原生的JSON格式数据，在管理页面数据处理是相当便利的，对于轻量化管理优势巨大，这也是选着MongoDB数据库最大的原因。更多的原因还有MongoDB支持对文档内嵌套的对象进行查询,前端可以方便实现一条查询就获得完整的数据。这类查询在关系型数据库难以实现。](#_Toc18891)

[4.3 WAF模块设计](#_Toc18891)

[4.3.1 正则检测模块与响应模块](#_Toc18891)

[正则检测模块是WAF系统的核心模块。主要功能提取访问中的请求信息与数据库中定义的规则是否有匹配部分再交由响应模块进行处理。如图4-2所示](#_Toc18891)

[](#_Toc18891)

[图4-2正则匹配模块结构图](#_Toc18891)

正则检测模块与响应模块整个访问中的工作流程图如下4-3所示。

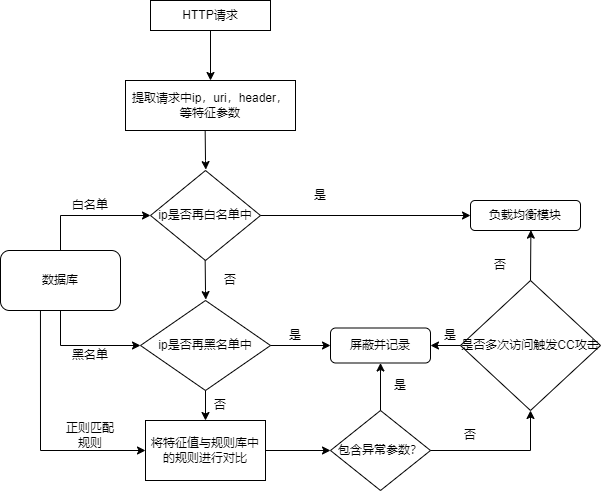


图4-3正则检测模块工作流程图

当一个HTTP/HTTPS请求到来时会被第一部分的参数特征提取模块进行解析，随后提取出在请求数据当中的一写特征参数、IP地址、URI还有请求参数等等特征参数内容。随后通过请求数据库返回的白名单信息中进行信息匹配，查看访问IP是否属于白名单IP，假如是白名单IP就直接调用负载均衡模块进行上游服务器分配，假如不在白名单列表中再判断是否在黑名单列表中，如果匹配成功交由响应模块进行处理，随后调用日志处理模块进行记录。当前两步骤顺利通过后就调用数据库返回的规则匹配库进行正则匹配，查看请求的数据是否有成功匹配规则的特征值。匹配成功则意味者这是一个恶意请求，将由响应模块进行处理，随后调用日志处理模块进行记录。成功通过这一步后，说明请求不是异常请求攻击，但是不排除存在CC攻击的可能，因此还需要查看最近该IP是否有多次访问的记录。假如确定是CC攻击那么将向客户端返回403，并且把访问记录通过日子管理系统发送到数据库的相关集合当中。当所有的流程都正常通过后才能够调用负载均衡模块进行后续工作。另当恶意访问触发时，设计的解决方案是再禁止了访问后，再返回一个网页告诉攻击者的行为已被发现，在一定程度上减轻攻击者的攻击欲望。

而无论访问能够顺利通过检测都会被日志管理模块进行收集记录。这一要求对于管理员分析数据十分重要。

4.3.2 日志处理分析模块

日志处理模块作为管理员分析攻击以及评估网页安全性的重要一环，我们对其的数据统计有一定的要求，首先我们需要分析HTTP请求的IP。管理可以根据IP的恶意访问历史将其加入黑名单,拦截后续请求,对已知的攻击源进行有效防御。可以评估IP的访问信誉,如果某IP长期有大量恶意访问记录,其信誉会很低,后续访问可以采取更严格的安全检测与防护措施等等。还要收集访问时间，访问时间可以让管理者由此预测WAF频繁受到攻击的可能时段，然后以此为标准加大防御力度。

除了以上比较重要的两点，还需要收集，请求的url、请求体、请求触发异常的规则、请求触发的攻击类型、服务器地址、用户访问所用的工具。而这些信息也将大大的提高管理者对于WAF的维护与评估。

4.3.4 负载均衡模块设计

因为WAF不具备给请求提供响应的功能，只会把安全的请求通过负载均衡的给上游的服务器集合进行均衡分配，而上游服务器处理完成后再把数据返回客户端，因此负载均衡就要求较低的响应时间与较高的吞吐量。

Nginx自带的负载均衡算法已经比较优秀，能够满足绝大多数场景的需要，也能够提供较高的响应速率。但是为了后续研究，本课题对优化的最小连接数负载军均衡算法进行研究。

首先算法在设计时考虑在动态时的复杂环境下，服务器的状态时不尽相同的，因此算法要在动态的环境下进行不断的变化。因此在动态下的情况下，加权最小连接法的解释就是，在为新的客户端请求分配服务器前，优先考虑当前连接请求数最小的服务器，这样能够保证各个服务器的运行效率更高，在动态的情况下，可以读取后端服务器实时连接数，再与原始权值进行对比，根据结果分配服务器。

核心思想即：把新的请求分配到上游服务器集群之中当在前任务连接数是最小的服务器。优点就是相比于静态的分析有着更好的负载均衡效果。缺点就是在较低的并发量时，会因为计算稍微提升响应时间。

算法描述是，设分配的服务器为S，设上游服务器节点为N，服务器用标号i（i=0,1,2,...，N-1），设为第i个服务器的初始权重，因为不同的服务器可能性能不同，因此根据服务器性能分配权重，设为第i个服务器的连接数，不同的阶段不同的服务器的连接数不同，这也是算法关键，由此得出新分配的服务器决定公式是

通过计算得到最小的连接权中比进行服务器分配就是最佳的服务器了。

4.4 WAF管理系统设计

WAF管理系统作为一个大块，其设计的目的最主要是为了方便管理人员对于WAF进行实时动态分析与观察，从而方便对WAF进行维护以及更新，能够应对更多的攻击。

4.4.1 数据展示模块

数据展示模块的主要是通过可视化的方式，向用户与管理员将WAF产生的各类数据通过图表、报表等形式进行直观展示,方便管理员分析与决策。主要信息如图4-4所示

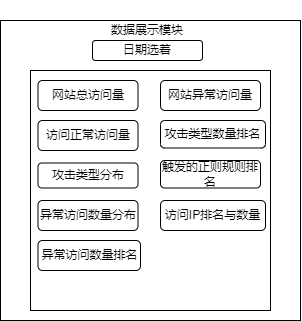


图4-4数据展示模块主要信息

网站的各种访问量展示能够直观反映系统的安全状况。如果异常访问量过高,或占比持续上升,则表明系统面临较大安全压力,需要关注。

访问量在各个时间段分布，管理员可以判断当前安全防护的不足,或预判未来可能遇到的安全问题,这有助于制定更加全面有效的安全策略。如果通过比较不同时期的访问统计发现异常访问或攻击活动有效降低,则可以判断管理员采取的策略或手段起到了效果,这也有助于后续制定更精准的控制措施。

各种攻击排名与触发的攻击规则排名，可以让管理员可以判断现有安全策略覆盖的不足,需要针对高频攻击采取更强有力的防护措施。攻击次数最高的攻击类型可判断为当前系统面临的高危威胁,管理员需要重点关注并采取有效对策。不同攻击类型及次数的变化可以评估系统整体风险度,如果高危攻击出现快速增长,则系统风险较大,面临较高威胁。

而访问的IP排行可以根据IP的历史访问记录评定其信誉,信誉低的IP访问可以接收更严格的安全检测与防护。对访问量过大的IP可以设置频率控制,限制其一定时间窗口内的最大访问次数,避免造成系统过载。TOP IP排行的变化可以反映出系统整体访问周期性,如每天的访问高峰期等。这为管理员制定对应访问期的流量控制提供参考。

4.4.2 日志管理模块

日志管理模块相对于数据展示模块，内容更加庞大与仔细，在这个模块当中能够查所有的访问日志，为了让管理者能够全面的分析WAF的各项性能，因此该模块需要展示最精细的内容，以及更准确的攻击类型分类，所有设计的过程需要更加有条理。

由于总的日志量非常巨大，并且页面的渲染能力有限，因此在日志信息渲染时需要根据攻击类型进行分类渲染，保证页面渲染的速度，以及服务器承载能力。

为管理者更方面的查看，需要设计时间查找条件与各种排序方式。只有这样管理员才能更好的通过

综上,日志管理是WAF管理系统的核心功能之一。完善的日志收集、存储与分析机制可以帮助管理员全面掌握WAF运行状态与安全事件,进行威胁识别与快速响应。日志分析WAF，管理与更新WAF，全面提高安全防护能力。日志管理是保障WAF管理系统安全运维的基石,其作用涉及安全监控、事件分析、系统管理等方方面面,这要求模块设计精细化与高度智能化,可以最大限度地发挥日志的作用

4.4.3 用户管理与权限控制模块

用户管理与权限模块的主要功能是实现管理系统的的登录用户的管理，以及角色新建、修改与角色分配。作为WAF的管理系统在权限上要求是非常严格的，注册功能只能由总管理员决定，用户修改也需要总管理员进行。

设计中这个页面中，分为用户管理与角色管理，角色亦为之用户能够查看的页面组成，而用户页面可以使得管理员了解各个用户的个人信息，假如用户由不良的行为能够直接得到个人信息，更快的追责个人。

两个页面都应该有搜索的功能，让管理员快速找到用户，新建与修改功能，快速的取消一个用户的权限与提升一个用户的权限，方面用户使用管理系统对WAF进行管理。

4.4.4 WAF管理模块

WAF管理模块涉及WAF中正则匹配与响应模块功能实现过程的数据的来源，因此在这个界面的权限比较重要，只有总管理员与web管理员才能够访问与使用。

在这个界面中管理员能够定义不同的攻击对应的匹配规则。还能够通过对日志中常出现的IP进行一个黑白名单的设置，保障WAF的安全性。

在这个页面中还应具备的功能有，规则的搜索、黑白名单的搜索、旧规则的删除与改进等等。只有各项功能齐全了，才能更好的管理与控制WAF的核心运转。

4.4.5.后端功能设计

本节讨论的后端功能是WAF管理系统的后端功能。因为在一定程度上WAF管理系统是与WAF本身分开的，课题考虑到，易于维护、部署方便等目标打算采用B/S架构设计WAF管理系统，B/S架构下只需要更新服务器端代码,所有的客户端都可以立即使用更新后的功能,无需逐个更新各个客户端。这大大降低了软件的维护成本与难度，对于B/S架构的应用,只需要将最新的程序部署到服务器端,所有的客户机可以立即使用,不需要在每台客户机上安装或部署相应的程序

为了能够后期带WAF管理系统进行更好的维护，本课题践行前后端分离的理念进行开发，前端只实现网页渲染的功能与一些简单逻辑，对于数据处理以及对数据库的增删改查等功能决定交由后端服务器解决。

后端服务器的选择，考虑到轻量化与维护的便捷化，决定后端服务器采用基于node.Js的express服务器。基于node.js的express具有强大的性能,特别适合开发数据密集型的 web 应用程序。express 可以很容易地集成主流的关系型数据库(如 MySQL)和文档数据库(如MongoDB)。这对于操作数据库的逻辑是非常便捷的。后端服务器需要实现的功能为，给WAF管理系统前端页面提供可调用的API接口，在前端页面需要完成某些功能时只要访问该接口即可，2、在服务器阶段进行数据的处理，处理后的数据存储到MongoDB数据库或者发送到前端页面。

1. Web应用防火墙测试与验证

本章首先对整个课题进行可行性分析随后针对WAF系统的各个模块进行功能验证以及性能测试，随后对WAF管理功能的各模块的功能进行测试。

6.1可行性分析

WAF的可行性分析需要考虑以下几个方面：

首先，WAF可以提高Web应用程序的安全性。随着Web应用程序的普及，网络攻击也越来越多。WAF可以检测和阻止恶意的HTTP流量，从而保护Web应用程序免受攻击。这对于那些需要保护用户数据和敏感信息的Web应用程序来说尤为重要。针对次问题本课题制作的WAF能够完成该目标

其次，WAF可以提高Web应用程序的可靠性。WAF可以检测和阻止恶意的HTTP流量，从而减少Web应用程序的崩溃和故障。这对于那些需要保持高可用性的Web应用程序来说尤为重要。针对这个问题，本课题使用的OpenResty是高性能的并且受广大开发人员所肯定，保证服务器的高可用性。

1. WAF可以提高Web应用程序的性能。WAF可以检测和阻止恶意的HTTP流量，从而减少Web应用程序的负载和响应时间。这对于那些需要保持高性能的Web应用程序来说尤为重要。针对这个问题，本课题使用了优化的负载均衡算法能够保证较低的响应时间。

综上所述，本课题实际实现的WAF有较大的实施可能，理论合理，符合实际

6.1 WAF系统各模块功能测试

本节针对WAF的核心模块进行实体测试，验证各模块的作用与实用性

6.1.1 规则匹配模块功能与响应模块测试

规则匹配模块依照数据库中的黑白名单集合以及正则规则集合对访问进行检测。因此对于模块的测试配置条件如下表6-1所示。

针对黑白名单的访问测试，设置规则一与规则二，针对针对URL的攻击制定规则三，通过post请求的攻击规则四。

|  |  |  |
| --- | --- | --- |
| 序号 | 触发条件 | 解释 |
| 规则一 | Black{IP：192.168.32.1} | 请求黑名单,IP为192.168.32.1时触发 |
| 规则二 | White{IP:192.168.32.4} | 请求白名单,IP为192.168.32.4时触发 |
| 规则三 | {url:”../”method:”get”} | 当请求方法是get方法时,并且请求Url中含有非法字符串”../”时触发 |
| 规则四 | {url:”(?:(union(.\*?)select))”method:”post”} | Post请求体中包含 "union" 和 "select" 之间的任意字符时触发 |

表6-1规则匹配模块测试条件

规则一很好理解,当访问的IP为设定的黑名单IP时会被阻隔,直接访问WAF代理的服务器http://192.168.127.3/a?a=../时候并且请求参数为a=../响应结果如图6-2所示,由此可知WAF能够拦截请求。

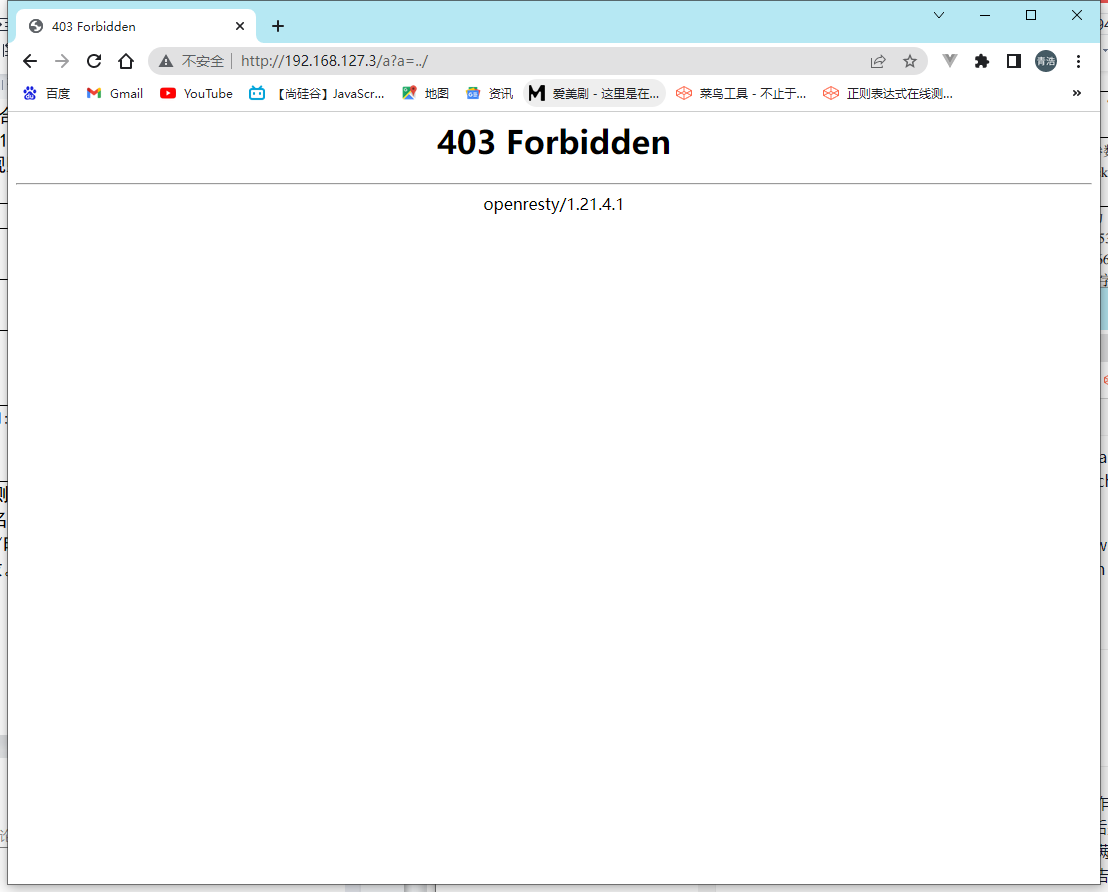


图6-2阻止黑名单访问结果图

规则二则表示为当访问的客户机IP是192.168.127.4时表示为白名单用户，可以不进行后边的所有测试，直接进入代理的网站，如图6-3所示

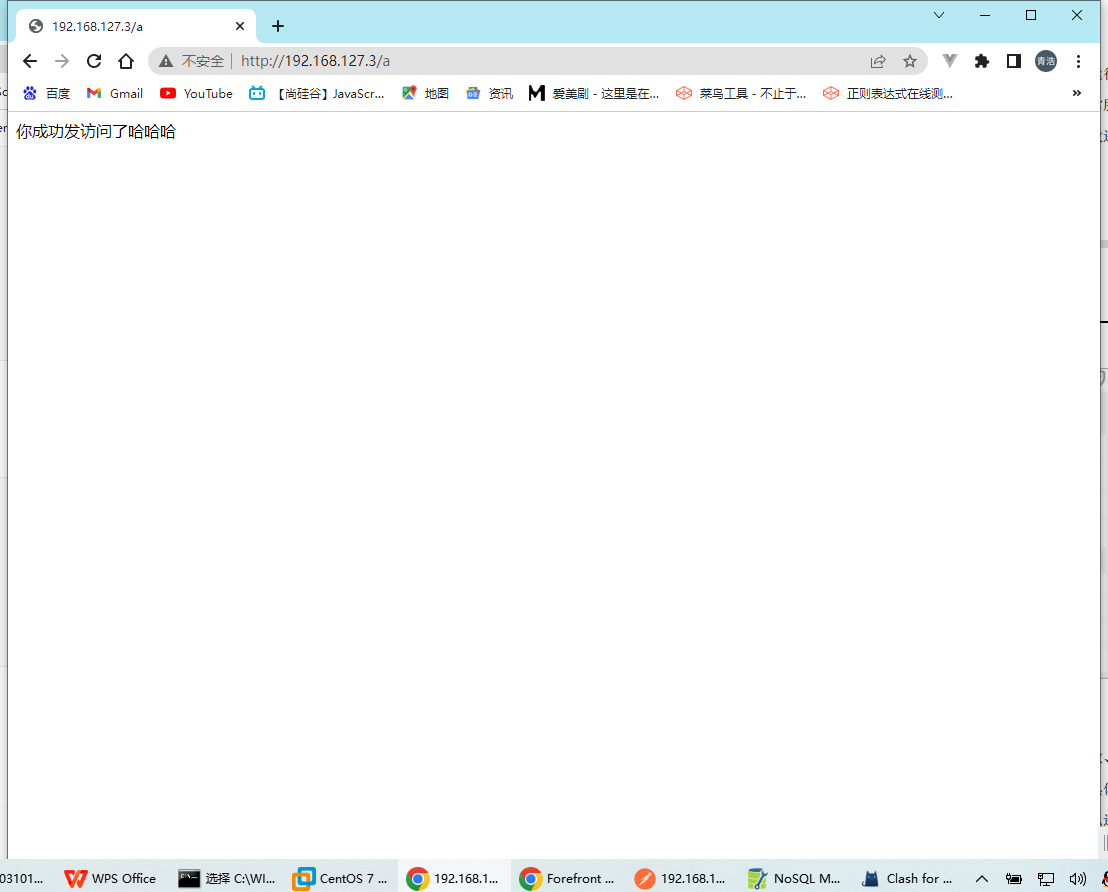


图6-3成功访问结果图

规则三是针对Url攻击的异常检测，当访问信息中携带了与正则匹配规则相符的字符串说明攻击成立，立即响应。返回内容如图6-4所示

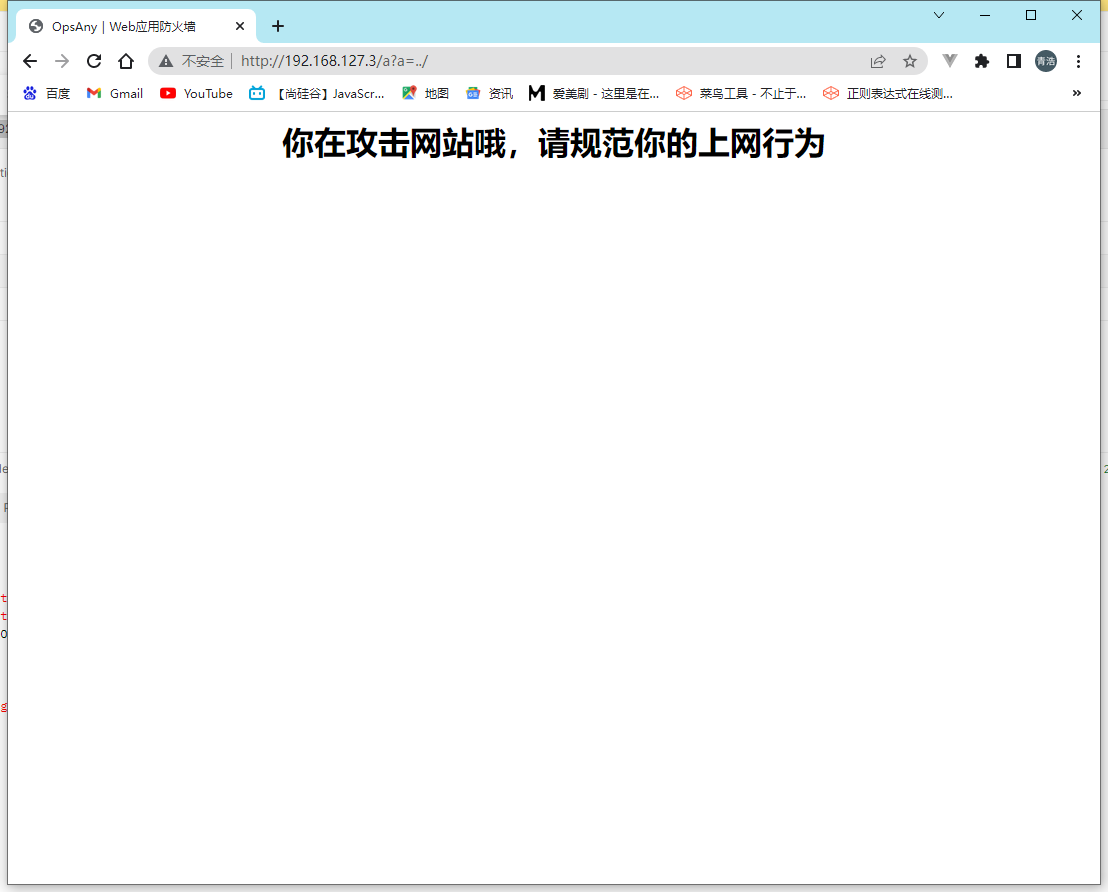


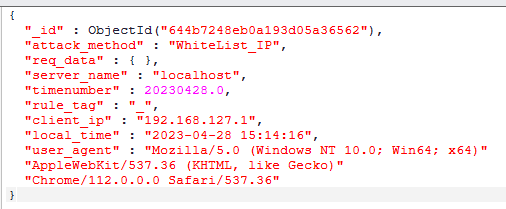
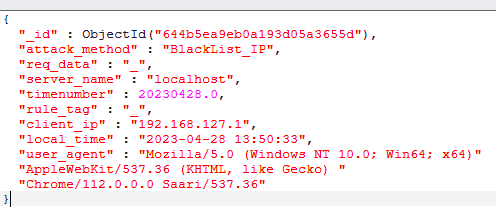
图6-4恶意攻击响应界面

规则四是针对post请求的异常检测，post请求与get请求存在一定差别，如url不能携带大量参数等，因此针对post进行专门检测，保证WAF保护能涉及各种请求，而相应如与图6-4相同。

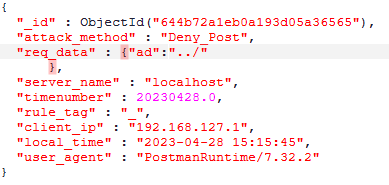
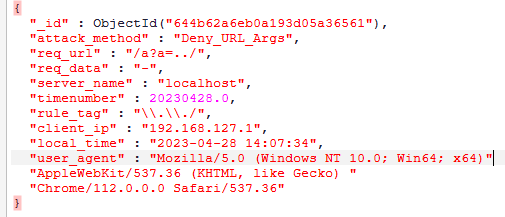
本课题WAF支持的各种攻击防护测试图可见附录，正文不再做过多复述。

6.1.3日志处理模块功能测试

日志处理模块主要提供日子的记录功能，该模块将访问的所有信息发送到数据库，供WAF管理系统使用。就上面4种规则的测试，通过MongoDB数据库可视化工具进行查看，分别得到了黑名单、白名单、URL攻击、POST攻击的日志收集图如6-5（a），6-5（b），6-5（c），6-5（d）



(a)黑名单访问日志 (b)白名单访问日志



(c)Url攻击日志 (d)基于Post攻击日志

图6-5各类日志

以黑名单访问日志为例进行解析，"attack\_method"：这是攻击的方法，这里是 "BlackList\_IP"，是指黑名单 IP 地址攻击。"req\_data"：这是请求数据，这里是 "\_"，没有提供任何有效的请求数据。"server\_name"：这是服务器名称，这里是 "localhost"，攻击发生在本地服务器上。"timenumber"：这是时间戳，这里是 20230428.0，表示攻击发生的日期和时间。"rule\_tag"：这是规则标签，这里是 "\_"，因为没有触发任何规则标签。"client\_ip"：这是客户端 IP 地址，这里是 "192.168.127.1"，是攻击者的 IP 地址。"local\_time"：这是本地时间，这里是 "2023-04-28 13:50:33"，表示攻击发生的本地日期和时间。"user\_agent"：这是用户代理，这里是 "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36"，表示攻击者使用的浏览器和操作系统。

6.2 系统管理模块测试

本节针对WAF管理的各个模块与按钮以及逻辑进行实体测试，验证各模块的作用与实用性。

1. 登陆页面

登录页面如下图6-6所示



(a)登录页面 (b)密码或者用户名错误

图6-6管理员登录页面

WAF管理员在登陆页面6-6(a)进行登录功能，成功够方可进入管理系统。当输入密码错误后与后端查询结果不匹配，会报错404状态如图6-6(b)所示

1. 系统主页面（数据展示模块）

登录成功后进入主页面由各种访问记录及其占比，各种异常访问时间分布以及排名、各种攻击排名排名、访问IP排名、触发异常正则排名组成。

[6.2.2日志展示模块测试 28](#_Toc6809)

[6.2.3 权限管理模块测试 28](#_Toc29887)

[6.2.4 Waf管理模块测试 29](#_Toc14233)

[6.3 系统整体防御效果测试 29](#_Toc14233)

[6.4 系统性能测试 29](#_Toc14233)

[6.5 系统测试结果分析 29](#_Toc14233)

[7 结论 36](#_Toc23700)

[谢 辞 37](#_Toc19019)

[参考文献 38](#_Toc22279)

[附 录 39](#_Toc32160)