

本科生毕业论文（设计）



题 目  基于代码混淆的Web站点动态防御系统

学 院  软件学院

专 业 软件工程

学生姓名 邵欣欣

学 号  **2015141463138** 年级 **2015**

指导教师 王海舟

教务处制表

二Ο一九年四月二十日

基于代码混淆的Web站点动态防御系统

软件工程

**学生** 邵欣欣 **指导老师** 王海舟

小型网站的安全问题没有受到应有的关注。很多企业的网上在线业务系统，会被机器人通过模拟人的行为发动攻击。导致的后果有抢占资源，冒用身份，欺诈交易，和信息泄露等。像自动化漏洞扫描、撞库扫号、恶意爬虫、暴力破解、薅羊毛等自动化攻击技术的发展速度丝毫不亚于企业新业务的发展，安全手段的乏力始终是企业IT的一个难题。当前市场上安全防护产品大多都是基于规则，产品总体偏重于匹配特征。传统基于特征和规则的入侵检测系统、Web应用防火墙（WAF）等已经无法有效地检测和拦截这些攻击，网站的防护急需一种新型的方式来应对。针对以上问题，本文设计开发了一个在Nginx反向代理环境下使用的Web防御模块，帮助网站开发者解决上述问题。

本文主要内容如下：

1. 分析目前应对自动化攻击的解决方案的优势和不足，结合各网站实际实施的安全防御现状，提出改进点。
2. 分析本系统需要实现的目标，以及实现思路。进行详细的需求分析；设计系统架构，提出应该使用的架构模式；将系统划分成数据采集模块和监控系统展示模块两个模块，选择每一个模块所采用的开发语言和开发框架等技术手段；设计系统的数据结构，完成数据库的数据字典设计。
3. 设计和实现如下的模块：通过动态ID匹配技术对客户端的请求进行过滤；通过浏览器插件对动态ID匹配机制在浏览器端提供支持；通过代码混淆技术对项目涉及到的JavaScript脚本和目标站点的JavaScript脚本做混淆处理。
4. 利用一个实际的网站做测试。从性能，容错来改进系统，保证不影响被防御站点的功能使用。

**关键字**：反向代理；Web防御；Nginx模块开发；动态ID

英文摘要

目录

第一章 绪论

1.1 研究背景及意义

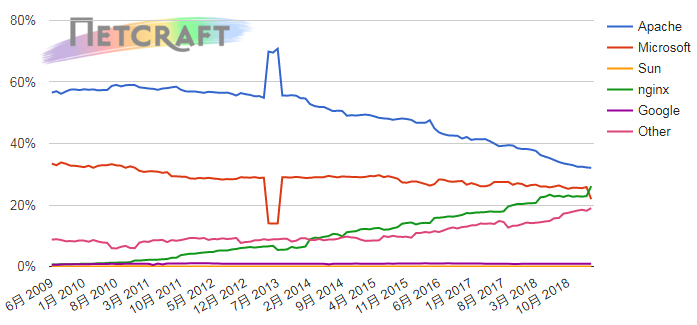
网站开发中，最初，新技术的使用通常以提供服务为着眼点，而忽略安全性，没有考虑用户的不可信。目前，网站的使用面临很多安全威胁。例如政府服务类网站，需要向用户提供查询服务，但是一旦遭遇爬虫，很容易被攻陷，或者消耗资源，或者被发现网站漏洞;电信运营商的网上营业厅，银行、甚至很多企业的网上在线业务系统，经常会被机器人通过模拟人的行为发动攻击。这些非“人”行为，极易得手的最根本原因是，面对看似完全正常的业务逻辑操作，现有安全手段却乏力于如何做出甄别，而这些威胁或者攻击一旦成功，即可抢占各类资源，进而可冒用身份，进行欺诈交易，甚至获得大量的客户信息、企业信息，造成大规模信息泄露。比如：针对一些商城的兑换服务，进行篡改积分，一些航空公司每每放出资源，几秒钟内就会被黄牛党抢光，抢占低价票资源。

当前市场上安全防护产品大多都是基于规则，产品总体偏重于匹配特征。显然，这样不停地找漏洞、打补丁令企业置身于千变万化攻击的被动中。特别是随着互联网+在国内的大力发展，像自动化漏洞扫描、撞库扫号、恶意爬虫、暴力破解、薅羊毛等自动化攻击技术的发展速度丝毫不亚于企业新业务的发展，安全手段的乏力始终是企业IT的一个难题。传统基于特征和规则的入侵检测系统、Web应用防火墙（WAF）等已经无法有效地检测和拦截这些攻击，网站的防护急需一种新型的方式来应对。

同时，我们知道，Javascript脚本是一种可读性较高的解释性语言，通常直接嵌入到网页文件中，一般用户都可以使用浏览器获得源码，对攻击者来说可轻易地获取源码进行逆向分析和恶意利用，这对承载Web应用重要核心业务流程信息的Javascript安全性带来巨大威胁。[1]

鉴于此，本项目拟通过人机识别技术和动态ID匹配技术对客户端的请求进行人机识别，以及通过随机化代码混淆技术对目标站点页面关键元素进行封装处理后再发送给用户，实现对网站的动态防御，有效降低自动化攻击技术的安全威胁，进一步防护网站的安全。

Nginx 是优秀的代理服务器，同时表现出众。图1-1，Netcraft发布的3月份调研数据



Web服务器开发人员：域的市场份额

1.2 研究现状

美国国家技术委员会在2011年提出“移动目标防御”(MTD)的概念，也有学者将MTD技术称为“动态防御技术”、“动态弹性安全防御技术”或者“动态赋能网络防御技术”。

动态防御不同于以往的网络安全研究思路，它旨在部署和运行不确定、随机动态的网络和系统，让攻击者难以发现目标。动态防御还可以主动欺骗攻击者，扰乱攻击者的视线，将具引入死胡同，至少可以设置一个伪目标/诱饵，诱骗攻击者对具实施攻击，从而触发攻击告警。动态防御改变了网络防御被动的态势，改变了攻防双方的“游戏规则”，真止实现“主动”防御。[6]

我国也对动态防御技术进行了重点研究，邬江兴院士提出了网络空问拟态防御CMD思想。CMD理论在可靠性领域非相似余度架构基础上导入异构冗余动态重构机制，造成在功能不变条件下，目标对象内部的非相似余度构造元素始终在作数量或类型、时问或空问维度上的策略性变化或变换，用不确定防御原理来对抗网络空问的确定或不确定威肋。[6]

在邬江兴院士文章中提到，拟态安全防御是指在主动和被动触发条件下动态地、伪随机地选择执行各种硬件变体以及相应的软件变体，使得内外部攻击者观察到的硬件执行环境和软件工作状况非常不确定，无法或很难构建起基于漏洞(bug)或后门的攻击链.以达成降低系统安全风险的目的。[5]

目前的国内研究以高效能计算为目标，从体系结构创新入手，对高性能计算在多个典型领域的应用、结构和效能关系进行了深入分析.揭示出“刚性不变的体系结构支持差异巨大的应用是使计算效能低下的根本原因”，引入了“应用决定结构，结构决定效能”的理念，提出了基于多维重构函数化结构与动态多变体运行机制的拟态计算体系—拟态计算(mimic computing，MC)。拟态计算固有的随机性、动态性和不确定性阻断了目前攻击技术所依赖的攻击链完整性。拟态计算的信息系统具备内在的主动防御能力,因此，基于拟态计算的信息系统具备内在的主动防御能力，称为拟态安全防御(mimic security defense, MSD)[5]

MSD主要针对网络空间攻击成本和防御成本的严重不对称性，以及我国信息领域核心技术与产业基础严重滞后国家安全需求的严峻性而提出，它是一种改变游戏规则的变革性技术.力图扭转目前网络空间“易攻难守”的战略格局。[5]

美国阿贡国家实验室2016年3月22日报道，更多的动态防御技术被美国国家专利局授权。2015年美国新型动态防御公司吸引到投资界的关注，动态防御概念的3家公司获得超过5000万美元的融资。具中，CrowdStrike(动态防御概念)获得3000万美元天使轮融资，Morphisec获得800万美元融资，ShapeSecurity获得2600万美元融资。北京卫达科一技有限公司将动态防御技术应用到内网防御、边界防御、工控防御、DDos防御、web应用防御、云安全等应用场景，研发6个系列的产品。[6]

截止目前，现有的网络安全产品大多采用传统的防火墙技术和入侵检测技术来实现对Web网站进行防护，在面对自动化攻击时十分脆弱。瑞数信息技术（上海）有限公司提供了类似的产品。

（1）动态算法生成：每次检查代码逻辑与形态均不同，且有效时间随整体访问量动态调整，攻击者每次必须在极短的时间内完成逆向，大幅提升了攻击难度，迫使攻击者放弃攻击。

（2）真实环境检查：利用不同浏览器中脚本引擎对特殊属性和语法的不同处理，验证真实的浏览器形态，大幅提升攻击者伪造终端运行环境的难度。

（3）攻击行为模式分析：基于人机互动理论，分析终端操作行为模式，有效识别非人为的操作行为，有效防止低频模拟操作攻击



图1- 3 瑞数信息公司动态安全

我们在此项目中将开发动态的防御，使防御手段变动，增强防御的机制，增加攻击者破解的难度，使用动态的id让攻击者无法用规律的自动化计算快速攻击、获取页面信息。

1.3 主要工作

Nginx是本项目的开发平台之一，通过阅读平台的使用指南，熟练使用此平台。然后，在Nginx的配置文件中修改服务器配置以及增加一个块来定义反向代理服务器。

该反向代理服务器接收客户端对目标服务器的请求，通过ID匹配和人机识别机制的配合，判定是否应向目标服务器发送客户端的请求。若向目标服务器发送了客户端的请求并接收了目标服务器返回的文件，再通过随机化技术封装目标服务器返回的文件后下发给客户端。

使用含有动态ID匹配设计的人机识别机制

在Nginx平台上搭建反向代理服务器后，反向代理服务器接收客户端发来的请求，如果请求没有携带可用于特定的人机验证的信息，进入人机验证流程。

通过了人机验证流程判断为“人”，才将客户端的请求发给目标服务器。否则，执行应对自动化的处理。

通过混淆技术对js脚本文件进行混淆。

在返回目标服务器的响应内容前， 对发给用户的文件进行混淆操作。使用JavaScript混淆技术对受保护站点JavaScript脚本和客户端ID生成算法进行混淆处理，增加攻击者对网页的逆向分析的成本。

网站运营商家运用此技术可以防止恶意爬虫、薅羊毛等自动化攻击，不仅可以减少因访问量降低和销量、广告收益降低的恶性循环导致的经济损失，还保障了用户信息安全。

1.4 论文组织

这篇论文内容结构如下：

第一章，。

第二章。

第三章。

第四章。

第五章。

第六章完总结整个项目，

1.5 小结

本章主要描述了基于代码混淆的Web站点动态防御系统系统的开发背景和开发意义，在分析研究了现有产品的研究现状和不足之处后，对本系统应该进行的改进进行了分析，进而提出了本系统应做的主要工作即功能需求，最后介绍了整篇论文的实现思路和组织结构。

第二章 预备知识

2.1开发技术

基于代码混淆的Web站点动态防御系统主要可以分成两个模块，即动态ID验证模块和使用混淆技术的代码保护模块。动态ID验证模块是基于前端浏览器插件和后端ID模块相互配合的B/S架构；使用混淆技术的代码保护模块是使用JavaScript代码混淆技术做安全保护。

1. Nginx模块开发

Nginx，一个高性能的HTTP和反向代理服务器，同时也是一个 IMAP/POP3/SMTP 代理服务器。

本系统使用了其核心subrequest模块，proxy模块来开发了一个第三方模块。运用Nginx基本数据结构，模块结构，事件处理方式，共享内存管理和引用库的技术等进行模块开发。

1. Gumbo

Gumbo 为开发人员提供一个轻量级的 HTML 解析库，它没有外部依赖，而且大多数语言都可调用它。Google 开放了Gumbo的源代码。该库符合HTML 解析算法标准，它通过了所有的html5lib-0.95测试，并且已经在 Google 索引的 25 亿个网页上进行过测试。

本系统使用Gumbo进行HTML解析来寻找改造目标网站页面内容的代码插入点。

1. Duktape

Duktape是一个轻量级的嵌入式 JavaScript 引擎，对ECMAScript兼容性较好。Duktape专注于可移植性和低占用率，可以被轻松地集成进一个 C/C++ 项目中。

本系统使用JavaScript程序代码实现混淆算法，来混淆受保护站点的JavaScript脚本文件。因此，使用Duktape执行实现混淆算法的JavaScript代码。

1. Chrome Extension

Chrome Extension扩展浏览器的功能，例如：捕捉特定网页的内容，捕捉HTTP报文，捕捉用户浏览动作，改变浏览器地址栏/起始页/书签/Tab等界面元素的行为，与别的站点通信，修改网页内容等。

Chrome Extension开发涉及到HTML，JavaScript和CSS技术点。浏览器扩展拥有一定高度的权限，在实现“捕获页面发出的HTTP请求请在url上缀加ID”这一目标过程中，使用浏览器扩展是开发变得清晰明朗，避开了包括解析HTML页面超链接，文件引用和JavaScript函数覆盖等一系列复杂的实现。

在本系统中，Nginx模块在转发目标站点的响应到浏览器的过程中向响应页面注入ID，然后浏览器扩展从页面捕捉到ID。实现了ID验证机制的前后端配合。

1. AugularJS

AngularJS 是一个 JavaScript 框架。它是一个以 JavaScript 编写的库。以一个 JavaScript 文件形式发布的，可通过 script 标签添加到网页中。它通过指令扩展了 HTML，且通过表达式绑定数据到 HTML。

在本系统的浏览器插件“protect”的开发中，使用了ng-app 指令定义一个 AngularJS 应用程序。ng-model 指令把元素值（比如输入域的值）绑定到应用程序。

2.2小结

本章主要描述了开发基于代码混淆的Web站点动态防御系统所需要的预备技术和知识。主要介绍使用到的Nginx项目知识，包括模块结构，如何引用库到Nginx项目中，以及引用的Gumbo库和Duktape库，并简单描述了基于代码混淆的Web站点动态防御系统如何将这些技术引用其中。

第三章 系统分析

3.1 需求分析

基于代码混淆的web站点动态防御系统的需求从前后端两个方面来分析

Chrome Extension “protect”能够接收并存储用户设置的要作用的网页的一类网址的一个通配-pattern；抓取浏览器中网址符合该pattern的网页页面中id属性为” hxID”的“input”元素的“value”属性获得ID值;截获浏览器发往网址符合pattern的HTTP请求，在添加ID后再前往该网址。

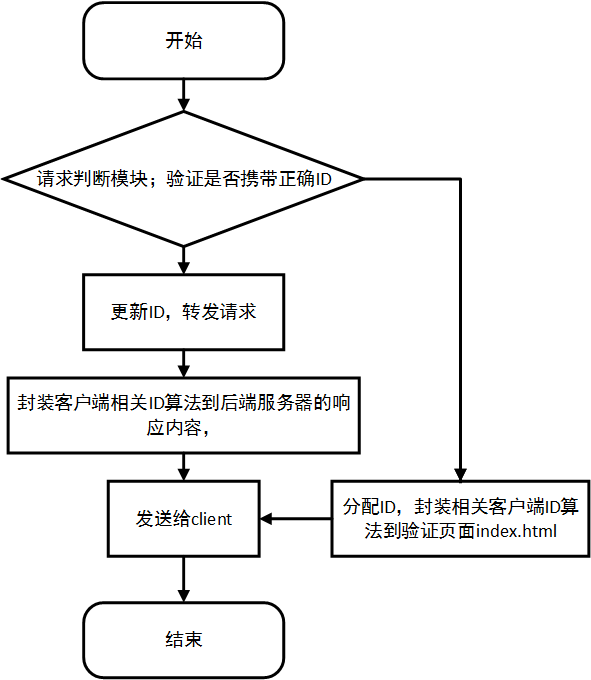
Nginx第三方模块ngx\_http\_ipslab\_module能够管理（IP，ID）对，包括给新访问新增（IP，ID），根据IP查找（IP，ID），更新（IP，ID），删除（IP，ID）；能够转发浏览器请求；转发目标服务器响应；能够向所转发的响应内容中注入ID相关信息；能够混淆JavaScript代码。

3.1.1 数据分析

基于代码混淆的web站点动态防御系统JavaWeb可以从Slab共享内存中的数据和内部交互，Chrome Extension 中的数据和内部交互以及二者之间的交互这三方面来进行分析。

1. Nginx模块ngx\_http\_ipslab\_module中的数据及内部交互

使用红黑树存储（IP，ID）对；节点数据结构以IP为键值，包括在红黑树结构上的增加，删除，查找，修改操作。



1. Chrome Extension 中的数据及内部交互

扩展程序储存用户数据，即用户输入的要作用的网址模式。在加载到相应网站的网页页面后，抓取携带在网页页面中的ID并临时存储，在用户操作使浏览器发起符合网址模式的请求时，依据ID生成新的newID，携带在请求中发出。

从当前网页页面中抓取ID，

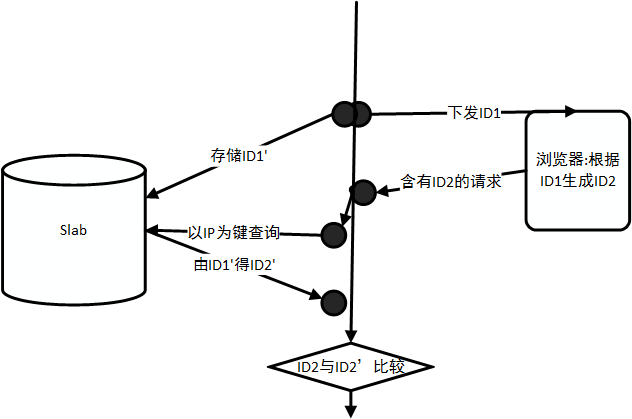
1. 前后端数据交互

对于针对某一个IP产生的ID，我们将它下发给浏览器，同时存储到Slab,为方便区分,分别称它们为ID1和ID1’。

当浏览器根据ID1生成ID2，再次发起请求时，携带ID2。

Nginx反向代理接收到请求，后根据IP从slab中找到ID1’,并随即生成ID2‘。

比较ID2与ID2‘来确定是否是合法请求。



3.1.2 功能分析

基于代码混淆的Web站点动态防御系统使用的用户是为自己的网站部署Nginx反向代理来增加安全，提高性能的网站技术人员。在选取Nginx提供的服务时，用户可以添加部署本模块来过滤掉一些自动化攻击，在一定程度上增加网站防御系数。为了实现过滤掉一些自动化攻击的功能，将基于代码混淆的Web站点动态防御系统主要分为两个部分：ID验证模块和混淆保护模块，其中ID验证模块又包括浏览器扩展和Nginx后端验证两部分相互配合来实现。

3.1.2.1 动态ID匹配进行人机识别

在slab共享内存中ID的更新，存储，修改，查看

浏览器与代理服务器之间ID交流的逻辑功能

3.1.2.2 混淆功能

在返回目标服务器的响应内容前， 对发给用户的文件进行混淆操作。混淆使用随机代码混淆技术对受保护站点页面关键元素和客户端ID生成算法进行随机化混淆处理，阻止攻击者对网页的逆向分析。

3.1.3 性能分析

基于代码混淆的Web站点动态防御系统，结合实际使用情况需要满足以下非功能性要求。首先要保证可用性和性能。

1. 一定的透明性

作为以反向代理为基础的系统，要保证用户使用受保护网站提供的服务时通畅的。保证其完整性，本系统开发的模块必须具有一定的透明性，对用户正常使用服务不造成影响。

1. 性能

保证一定的性能，不能在反向代理处做无故的滞留影响响应速度。具体度量不超过原目标站点响应时间的两倍。

1. 容错性

基于代码混淆的Web站点动态防御系统，对一些开发中无法预料的请求可能会在使用中出现并影响到用户正常使用的情况，需提供一种容错的补充方案，一定保证受保护站点业务服务可以提供给用户。

1. 可用性

对于供浏览器用户使用的Chrome Extension“protect”需要简洁清楚的界面，需要给出操作成功的提示信息和错误操作的出错详细信息，对用户有指导作用，提供给用户的步骤要简单单调，提高可用性。

对于供网站部署人员，本系统提供的Nginx第三方模块ngx\_http\_ipslab\_modul是通过指令配置的方法来使用。为网站部署人员提供详细的指令介绍，配合在日志中输出详细的操作信息来让网站部署人员更深入了解本系统。

3.2 系统设计

基于代码混淆的Web站点动态防御系统。

3.2.1 架构设计

* 1. 客户端算法：在发起请求时，使用之前服务器下发的ID生成新的newID,并附加newID到请求中。
  2. 服务器端算法流程：

1. 存储下发给IP的ID即（IP，ID）记录。
2. 接收请求，进入自动化请求过滤环节。通过进入④，未通过进入③。
3. 对于ID不匹配的IP请求更新（IP，ID）记录，对于没有ID记录的IP请求添加（IP，ID）记录到存储库中，不转发用户请求到第三方服务器，同时响应用户一个特殊构造的HTML页面，该HTML页面保留了用户未通过的请求信息以及ID相关内容，以便用户再次发起携带ID的请求。

向第三方服务器发起请求，得到响应后，注入“检测自动化攻击”模块客户端需要的代码并对网页代码做保护后，转发给用户，更新（IP，ID）记录。：

图3-3 系统架构图

JavaWeb项目监控系统由采集模块和监控展示模块构成，数据库选择了MySQL，服务器选择了Tomcat8，系统的部署图如下：

图3-4 系统部署图

3.2.2 数据设计

ngx\_http\_ipslab\_module模块内的结构体设计如下：

表3-1 ngx\_http\_ipslab\_module模块配置结构体ngx\_http\_ipslab\_conf\_t

|  |  |  |
| --- | --- | --- |
| **变量名称** | **变量类型** | **含义** |
| shmsize\_int | ngx\_int\_t | 共享内存的大小，以pagesize=4096为单位大小 |
| shpool | ngx\_slab\_pool\_t | 共享内存结构体 |
| sh | ngx\_http\_ipslab\_shm\_t | 控制红黑树和链表 |

表3-2 存储(IP,ID)对的节点ngx\_http\_ipslab\_node\_t结构体

|  |  |  |
| --- | --- | --- |
| **变量名称** | **变量类型** | **含义** |
| rbtree\_node\_data | u\_char | 实际存储信息开始的位置 |
| slab\_ID | u\_char[33] | 下发给浏览器端的ID |
| last\_slab\_ID | u\_char[33] | 上一次下发给浏览器端的ID |
| queue | ngx\_queue\_t | 一条一定长度保持访问记录的链表，用来淘汰过期节点 |

表3-3 ngx\_http\_ipslab\_shm\_t结构体

|  |  |  |
| --- | --- | --- |
| **字段名称** | **存储类型** | **含义** |
| rbtree | ngx\_rbtree\_t | 红黑树用于快速检索 |
| sentinel | ngx\_rbtree\_node\_t | 使用红黑树必须定义的哨兵节点 |
| queue | ngx\_queue\_t | 淘汰链表 |

表3-4 ngx\_http\_ipslab\_ctx\_t表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **存储类型** | **含义** |
| out\_ctx | ngx\_chain\_t \* | 传递子请求返回结果 |
| hxID | ngx\_str\_t | 父子请求间传递ID |

枚举类型Append\_flag定义

|  |  |  |
| --- | --- | --- |
| **字段名称** | **存储类型** | **含义** |
| pure\_chrctr | 0 | 标识添加“hxID=xxxx” |
| ampersand\_chrctr | 1 | 标识添加“&hxID=xxxx” |
| question\_chrctr | 2 | 标识添加“?hxID=xxxx” |

**3.2.3 UI设计**

基于代码混淆的Web站点动态防御系统中，Nginx端模块的使用实在Nginx的启动配置文件nginx.conf文件中配置使用。浏览器端的浏览器扩展页面设计简介，有必要提示信息，有操作成功提示，以下是浏览器扩展的界面设计：

图3-6 监控系统首页

3.3 小结

本章节着重分析了基于代码混淆的Web站点动态防御系统的功能性需求以及性能需求，从三个组件描述了系统的设计实现，详细描述了ngx\_http\_ipslab\_module模块所使用到的数据结构。

第四章 系统实现

4.1系统环境

基于代码混淆的Web站点动态防御系统

本项目将ID验证模块服务器端支持与混淆模块在ngx\_http\_ipslab\_module中用C实现。

将ID验证模块浏览器端支持使用Chrome Extension实现，使用HTML和JavaScript语言。

本项目开发环境：

在Windows操作系统下，安装Cygwin进行Nginx编译，部署。并将项目导入Eclipse-cdt IDE进行开发。

4.2模块描述

JavaWeb项目监控系统的设计使用了模块化的方式，模块化的设计方式可以减少系统的耦合性，增加内聚程度，符合软件工程的标准。在JavaWeb项目监控系统中，系统可以被划分成两个模块：即数据采集模块和监控系统展示模块。数据采集模块负责采集用户网站的监控指标数据，具体包括网站接口性能即接口耗时的采集、服务器性能的采集即用户网站服务器CPU、内存、磁盘利用率的采集、网站用户参数的采集、以及采集过后的报警功能即当用户网站接口耗时或服务器资源使用率超过了用户预先设定的阈值将会报警的功能；监控系统展示模块以Web页面的形式输出，即将上述数据采集模块采集出的数据通过网站的形式展示给用户，具体的功能包括用户网站接口耗时查看功能、用户网站所在服务器资源即CPU、内存、磁盘等资源查看功能、用户网站历史参数的查看和和多种格式的下载功能、新建数据接入和已有数据接入的查看功能、用户对JavaWeb项目的配置的增删改查等功能[16]，以及用户登录注册和对个人信息的查看和修改等功能。系统的总体模块结构图如下：

图4-1 系统模块结构图

4.2.1 ID验证模块浏览器端支持

采用Chrome Extension，从网页页面获取ID生成newID，截获从网页发出的HTTP请求进行添加newID操作，然后再使用重定向技术真正发出请求。

Popup模块接收用户设置的网址模式：pattern并将其通过变量传递给Bcackground模块。

Content-Script模块保持从页面中抓取到的ID：hxID并将其通过message传递给background模块。

Background模块将收到的pattern和hxID 均存储到Chrome.storage中。

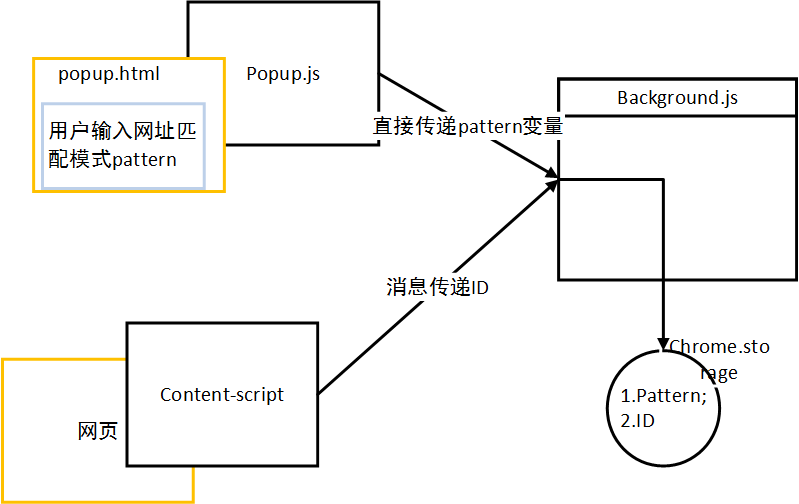
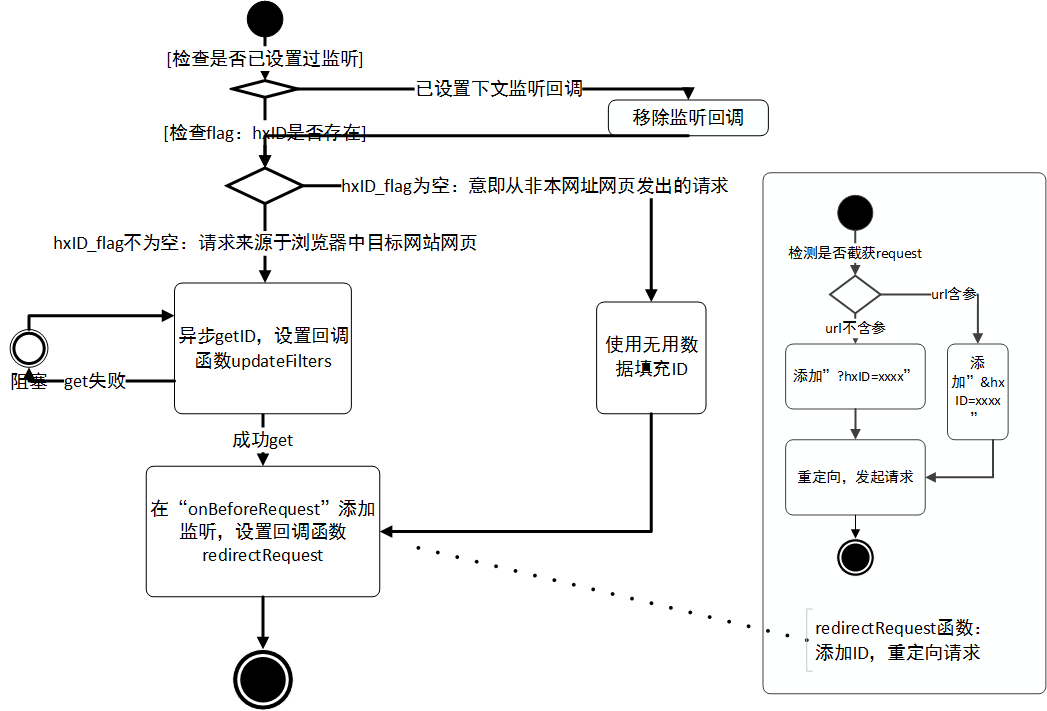


图4-2 系统模块结构图

Background.js的内部:



#### 4.2.1.1

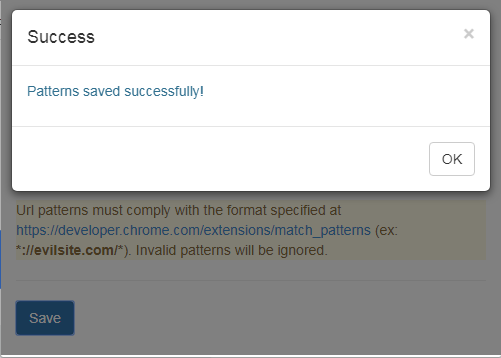
Nginx第三方模块ngx\_http\_ipslab\_module

#### s4.2.2.4代码混淆

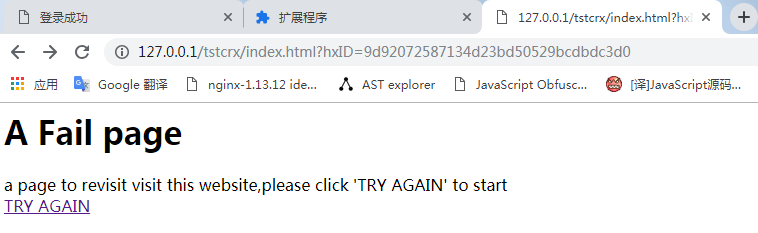
JavaWeb项目监控系统同样提供了类似于其它系统的账户管理功能，包括用户登录、注册、登出以及查看和修改账户信息等功能。

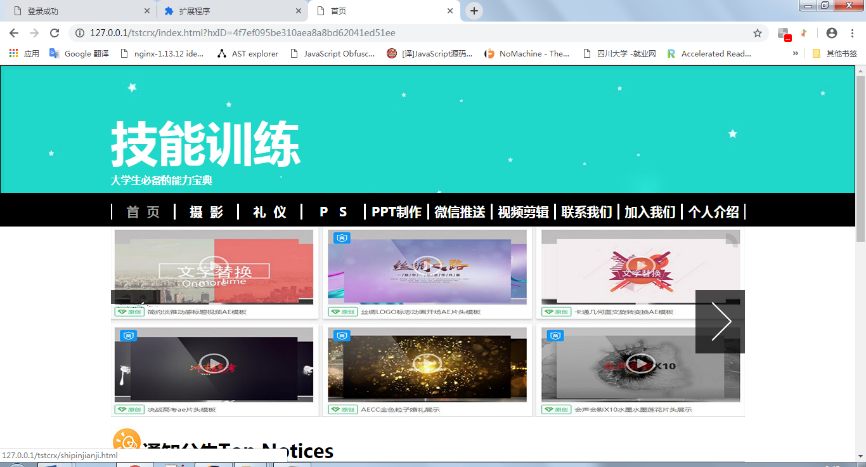
向

4.3运行效果











4.4小结

第三章描述了基于代码混淆的Web站点动态防御系统的概要的设计，本章对这些设计部分详细揭示了其中的细节，详细的描绘了系统的模块构成，并对每一个模块的版块组成和流程进行了阐述，描绘了其中具体的实现思路和技术，给出了一些板块的运行效果页面。

第五章 系统测试

系统测试可以保证一个软件系统稳定、可靠的运行，对JavaWeb项目监控系统进行系统测试可以保证系统满足用户的需求并且有着较高的质量，应该保证系统的bug率在允许的情况内[20]，JavaWeb项目监控系统的系统测试全程由四川大学软件学院崔博文完成。

5.1 测试环境

基于代码混淆的Web站点动态防御系统的测试：

在Windows 7 PC上安装并启动Nginx Windows版本服务器作为目标服务器，配置监听8080端口，部署的网站是“大学生技能训练网站”。这时，目标网站“大学生技能训练网站”，就可以通过”127.0.0.1:8080”访问到。

在Windows 7 PC上安装Cygwin 64 虚拟Linux包环境，通过该环境编译Nginx 1.12.2，编译时将“基于代码混淆的Web站点动态防御系统”开发的ngx\_http\_ipslab\_module模块编译进去。在nginx.conf文件中根据“用户手册进行配置”。

测试时Nginx-1.12.1的nginx.conf需要的配置有：

使用ngx\_http\_ipslab\_module模块自定义指令配置slab共享内存 8\*pagesize(4096)；

配置proxy\_buffer\_size 125K；

配置Keepalive\_time 0；

配置监听80端口；

配置符合”/tmpdir/”的location块代理到目标服务器8080端口。

具体在nginx.conf文件的对应的指令分别如下：

ip\_slab 8;

proxy\_buffer\_size 125k;

keepalive\_timeout 0;

listen 80;

location /tmpdir/{proxy\_pass 'http://127.0.0.1:8080//';}

指令在文件中的具体配置详细参见“用户指导手册”

5.2 测试内容

JavaWeb项目监控系统的系统测试采用的测试方法主要是黑盒测试，某些功能模块采用了黑盒与白盒二者相结合的方法[21]，将系统区分成多个模块，再对每个模块中的功能进行正确功能性测试，通过对期望输出和实际输出的比较，确定测试结果。

JavaWeb项目监控系统测试需要划分成两个模块：数据采集模块、监控系统展示模块。

下面是数据采集模块测试记录：

表5-1 测试01过程记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试编号** | 01 | | | |
| **测试模块** | 数据采集模块 | | | |
| **测试环境** | 计算机操作系统：Windows10，浏览器：Chrome | | | |
| **限制条件** | 无 | | | |
| **测试功能** | **输入内容** | **测试过程** | **输出期望** | **实际输出内容** |
| 用户网站接口耗时采集功能 | 用户给自己的网站配置数据采集模块，且有接口调用 | 将数据采集模块jar包导入用户网站，启动网站，观察耗时数据是否存储到了数据库中 | 用户网站接口耗时数据应该正确存储到数据库中 | 用户网站接口耗时数据能够正确存储到数据库中 |