**硕士学位论文**

(工程硕士)

基于页面内容分析的钓鱼网站检测系统的设计与实现

**DESIGN AND IMPLEMENTATION OF PHISHING WEBSITE DETECTION SYSTEM BASED ON THE CONTENT OF THE PAGE**

**王文帮**

**哈尔滨工业大学**

**2015 年 6 月**

国内图书分类号：TP311.5 学校代码：10213

国际图书分类号：621.3 密级：公开

**工程硕士学位论文**

基于页面内容分析的钓鱼网站检测系统的设计与实现

|  |  |  |
| --- | --- | --- |
| 硕士研究 Th | ： | 王文帮 |
| 导 师 | ： | 李 东 教授 |
| 副 导 师 | ： | 葛瑞海 助理研究员 |
| 申 请 学 位 | ： | 工程硕士 |
| 学 科 | ： | 软件工程 |
| 所 在 单 位 | ： | 软件学院 |
| 答 辩 日 期 | ： | 2015 年 6 月 |
| 授予学位单位 | ： | 哈尔滨工业大学 |

Classified Index：TP311 U.D.C.: 621.3

Dissertation for the Master’s Degree in Engineering

**DESIGN AND IMPLEMENTATION OF PHISHING WEBSITE DETECTION SYSTEM BASED ON THE CONTENT OF THE PAGE**

|  |  |
| --- | --- |
| **Candidate：** | WangWenBang |
| **Supervisor：** | Professor LiDong |
| **Associate Supervisor:** | Assistant Engineer GeRuiHai |
| **Academic Degree Applied for：** | Master of Engineering |
| **Speciality：** | Software Engineering |
| **Affiliation：** | School of Software |
| **Date of Defence：** | June, 2015 |
| **Degree-Conferring-Institution：** | Harbin Institute of Technology |

摘 要

哈尔滨工业大学工程硕士学位论文

随着网络层在人民生活中的快速发展，国家在网络安全监管方面上的需求变得越来越迫切。钓鱼网站检测系统是北京赛思信安技术股份有限公司的重点网络安全技术发展建设的项目之一。其中基于页面内容分析的钓鱼网站检测系统作为整体系统中的重要部分，其设计目标是要建立一个能够有效地检测出针对合法网站进行仿冒的钓鱼网站的处理系统。

论文以调研目前业界钓鱼网站的现状及发展成果为基础，介绍了国内和国外钓鱼网站检测的现状，并结合当前国家针对钓鱼网站的防御和检测情况，总结了目前国家对钓鱼网站的检测的要求和标准。通过全面且深入的分析来设计面向需求的钓鱼网站检测系统并加以实现。

论文以LINUX系统为开发平台，主要的开发语言采用C++并辅以java语言执行数据管理。系统组成部分包括URL去重模块、白名单匹配模块、多模匹配模块、哈希相似度判定模块、日志模块等模块。由模块间的协作实现新增URL条目的检测业务、汇报任务、本地日志生成等任务，满足了系统对钓鱼网站检测的要求。本文通过连接池的技术解决异步域名解析和下载HTML页面内容的工作，采用多模匹配技术解决了特征文本的筛选的困难，使用哈希算法进行相似的的匹配和判断。

该钓鱼网站检测系统可以提供快速、稳定的钓鱼网站检测服务。能够根据导入的新增URL条目，实现相应的检测任务。通过对该系统的不同层次的测试，结果可以通过基于页面内容分析的钓鱼网站检测系统的业务需求。基本达到功能性需求，在性能方面初步达到标准。

关键词：钓鱼网站；页面内容；多模匹配；哈希相似度

Abstract

哈尔滨工业大学工程硕士学位论文

With the rapid development of network layer in people's life, the demand of the state in the network security supervision becomes more and more urgent. Phishing detection system is Seth principal Technology Co., Ltd. Beijing focus on network security technology development and construction of the project. The analysis based on page content of phishing web detection system as an important part of the whole system and its design goal is to establish a can effectively detect the fake phishing sites processing system for legitimate sites.

The research at present industry phishing sites of the status and the achievements of development as the foundation, introduces the status of domestic and foreign fishing website detection, and combined with the current state for defense and inspection situation of phishing sites, summed up the requirements for the detection of phishing sites of our country. With comprehensive and detailed analysis to construct the phishing website detection system meets the overall system requirements.

The paper takes the LINUX system as the platform, C++ language used as the main development language and the Java language executes data management. The system components include URL to weight module, the white list matching module, the multi-mode matching module, the Hashi similarity judgment module, the log module and so on. From the collaboration between the modules to achieve the new URL detection business, reporting tasks, local log generation and other tasks, to meet the system requirements for phishing sites detection. The connection pool technology to solve the asynchronous DNS and downloads the HTML page content, a multi-mode matching technology to solve the difficulty of text feature selection, using a hashing algorithm of similarity matching and judgment.

The phishing site detection system can provide fast and stable phishing site detection service. According to the new URL entry, we can achieve the corresponding detection tasks. Through the test of different levels of the system, the results can be based on the content of the phishing website detection system business needs. Our system is basically able to meet the functional requirements and the performance of the initial reach of the standard.

Keywords: Phishing sites; Page content; Multi mode matching; Hash similarity

目 录

哈尔滨工业大学工程硕士学位论文

目 录

[摘 要](#_Toc686774360) 4

[Abstract](#_Toc686774361) 4

[第 5 章 系统的测试和分析 61](#_Toc686774362) 5

[5.1 测试环境 61](#_Toc686774363) 5

[5.2 单元测试 61](#_Toc686774364) 5

[5.3 集成测试 63](#_Toc686774365) 5

[5.4 系统测试 63](#_Toc686774366) 5

[5.5 本章小结 65](#_Toc686774367) 5

[第1章绪论](#_Toc686774368) 5

[1.1 课题来源及研究意义](#_Toc686774369) 5

[1.2 与课题相关的国内外研究综述](#_Toc686774370) 6

[1.3 本论文的主要工作内容](#_Toc686774371) 7

[第2章系统需求与关键技术分析](#_Toc686774372) 7

[2.1 需求分析](#_Toc686774373) 7

[2.2 关键技术分析](#_Toc686774374) 9

[2.3 本章小结](#_Toc686774375) 11

[第3章系统分析与设计](#_Toc686774376) 11

[3.1 系统总体设计](#_Toc686774377) 11

[3.2 系统功能模块设计](#_Toc686774378) 12

[3.3 本章小结](#_Toc686774379) 18

[第4章系统的实现](#_Toc686774380) 19

[4.1 URL去重模块实现](#_Toc686774381) 19

[4.2 白名单过滤模块实现](#_Toc686774382) 20

[4.3 连接池模块实现](#_Toc686774383) 20

[4.4 页面内容解析模块实现](#_Toc686774384) 21

[4.5 多模匹配模块实现](#_Toc686774385) 22

[4.6 正则表达式匹配模块实现](#_Toc686774386) 22

[4.7 哈希相似度检测模块实现](#_Toc686774387) 24

[4.8 日志模块实现](#_Toc686774388) 24

[4.9 本章小结](#_Toc686774389) 25

[第5章系统的测试和分析](#_Toc686774390) 25

[5.1 测试环境](#_Toc686774391) 25

[5.2 单元测试](#_Toc686774392) 25

[5.3 集成测试](#_Toc686774393) 26

[5.4 系统测试](#_Toc686774394) 26

[5.5 本章小结](#_Toc686774395) 28

[结论](#_Toc686774396) 28

[参考文献](#_Toc686774397) 28

3.2.8 日志模块设计 43

哈尔滨工业大学工程硕士学位论文

3.3 本章小结 44

第 4 章 系统的实现 45

4.1 URL 去重模块实现 45

4.2 白名单过滤模块实现 47

4.3 连接池模块实现 48

4.4 页面内容解析模块实现 51

4.5 多模匹配模块实现 53

4.6 正则表达式匹配模块实现 55

4.7 哈希相似度检测模块实现 57

4.8 日志模块实现 58

4.9 本章小结 60

# 第 5 章 系统的测试和分析 61

## 5.1 测试环境 61

## 5.2 单元测试 61

### 5.2.1 接口测试 61

### 5.2.2 局部数据结构测试 62

## 5.3 集成测试 63

## 5.4 系统测试 63

## 5.5 本章小结 65

结 论 67

参考文献 68

哈尔滨工业大学学位论文原创性声明及使用授权说明 71

致 谢 72

个人简历 73

# 第1章绪论

哈尔滨工业大学工程硕士学位论文

## 1.1 课题来源及研究意义

本课题来源于北京赛思信安技术股份有限公司与中科院信息工程研究所钓鱼网站的检测项目中的基于页面内容分析的钓鱼网站检测系统的设计与实现。针对当前有着较多的钓鱼网站通过深度模仿正规网站，诱导用户注册或登

录，通过采集敏感个人信息等使受害者蒙受经济损失。尤其在电子商务中有着很多的相关案例，使得消费群体面临着非常严峻的信任危机。另外，钓鱼网站的制作成本相对较小，借由正规网站版本更新速度慢的特点，只需要做出一版相似度很大的版本，就可以为很多不法运营商所使用，这使得网络安全环境受到了反复的极大冲击。

由此可见，对网络钓鱼的检测防御研究迫在眉睫，尤其是面临大量实时更新着的数据和复杂高仿冒网站时，对检测而言问题显得尤为突出。本系统采用了最直观的检测方式，由页面内容的分析入手，分解出各种检测项作为评价的标准，结合黑白名单库、whois信息参考等，通过综合各部分检测结果的汇总力争来检测出较为准确的结果。

## 1.2 与课题相关的国内外研究综述

在网络普及的当下，无论国内外，网络钓鱼攻击[1, 2]早已受到广泛重视，各国也逐步建立了反网络钓鱼联盟及类似的组织和团体，如APWG(Anti-Phishing Working Group), APAC(Anti-Phishing Alliance of China)等。

其中APAC，即为中国反钓鱼网站联盟，成立于2008年，由国内银行证券机构、域名注册服务机构、域名注册管理机构、电子商务网站等权威机构和专家学者组成的，是国内唯一权威的具有指导性价值的反钓鱼组织。

而APWG，成立于2003年，是全球专注于统一的全球应对网络犯罪通过发展的数据资源、数据标准和模型响应系统的工业、执法和政府联盟，汇集了企业受到网络钓鱼攻击，安全产品和服务公司、执法机构、政府机构、行业协会、

地区国际条约组织和通信公司。APWG有超过三千个成员来自世界各地的一千七百多家公司和机构，成员公司包括BitDefender, Symantec, VeriSign等，金融行业成员包括荷兰国际集团，维萨公司和美国银行家协会。

哈尔滨工业大学工程硕士学位论文

### 1.2.1 国外钓鱼网站检测发展及其现状

针对网络钓鱼的检测[3]研究开展的比较早，随着电子商务逐步融入人们的生活和新技术的开发，越来越多的检测手段和技术正在解决网络钓鱼所带给我们的困扰。

#### 1.2.1.1 基于URL的检测技术

1. URL黑白名单匹配检测

基于URL检测[4-13]是指通过检测网页的URL，就能区别当前所访问的网站是否是钓鱼网站。当前反钓鱼网站浏览器工具栏插件是最受欢迎的产品，可以广泛部署解决方案打击网络钓鱼网站，大部分的工具栏使用着黑白名单作为检测的数据库名单，方便维护和管理。通过查看黑名单/白名单过滤(B/W)数据库确定URL。检查结果将被交付用户与警告用户可能的钓鱼网站，或通报网站是合法的。尽管URL通过黑白名单匹配的检测技术实现容易而且命中率较为可观，但是却很难发现不在名单中的钓鱼网站是否是合法的，且确认黑名单需要大量动用人工力量去验证，效果经费投入比很低。通过简单的匹配、对比实现对网页的检测来分析结果，黑名单的获取和确定需要人工进行审核和校对，具有一定时间上的滞后性，而钓鱼网站的存活期相对较短，而且变化很快，因此这类方法并未在实时性方面上解决网络钓鱼攻击带来的问题。

2. 机器学习检测

应用机器学习方法的URL检测技术是直接利用URL检测钓鱼网站[14]，主要选择钓鱼网站页面中的URL作为特征向量生成训练数据，并采用分类器模型和应用分类器，相结合的方法分类数据信息。

##### 1) Garera算法

Garera等人通过分析钓鱼网站页面代码中的URL树的结构，详细的介绍了向量特征集的选取方法和过程，特征集由HTML网页的特征、域名特征、数据

类型的特征等共十多个元素构成，利用回归滤波器分类URL[15]。

哈尔滨工业大学工程硕士学位论文

由钓鱼网站目标URL的结构进行结构性辨析得出以下四种类型的URL字符串组成。HTML页面代码中的特征需要借助访问搜索引擎并通过页面渲染技术来提取，选则页面中URL条目链接的热度排行、爬取的页面在总页面库中的统计数据库、页面索引条目数据、HTML网页代码质量评价共6个特征。通过大量钓鱼网站的训练之后进行测试得出分类精度和准确率为95%-97%，

##### 2) Ma算法

采用Ma算法分析URL字符串中存在的可疑词汇和host属性，采用词语袋模型(bag-of-words)用作页面内容特征的选取，使用这个特征模型选取到的特征总量是十分可观的。经过调研，主流的检测特征中包含URL信息长度、hostname长度等代表性的特征。Host主机特征中可选取的向量特征信息可以选择whois属性、IP属性、domain属性等进行参考。

对于二元特征向量组的构建中，其中一组向量采用的是URL字符串中代表主机的符号，另一组向量则使用URL字符串中代表路径的符号作为选取标准。在建立这组二维向量组的过程中使用词语袋模型(bag-of-words)进行行为上的组成指导。

对于批量学习，Ma等专家通过分析了支持向量机的模型和算法，即SVM算法、朴素beyes算法等分类算法；通过on-line学习研究了被动贪婪算法、随机梯度降落回归滤波、感知器等算法的分类性能[16]。通过批量分类的算法取得了误差百分之五以下的准确度，分类精度在在线算法测试执行结果中近乎完美。

#### 1.2.1.2 基于启发式融合多特征的检测技术

2007年来自Carnegie Mellon University的Lorrie Cranor等人提出一种新颖的基于页面内容的检测方案，称为“CANTINA”，CANTINA[17]通过检测web页面的内容来确定其是否合法，利用已知的TF-IDF（逆文档频率）信息检索算法[18]进行检测，TF-IDF用于从大型语料库中提取信息进行比较和分类，利用TF-IDF的一种应用称为“Robust Hyperlinks”来克服坏链接[19]的问题。给定一个网页，计算页面上每一词汇的TF-IDF值，选取比例最高的5个词汇生成词法签名，放入搜索引擎中进行搜索，搜索结果如果与当前web页面的域名相匹配

则认为这是一个合法的网站，否则为钓鱼网站。

哈尔滨工业大学工程硕士学位论文

#### 1.2.1.3 基于视觉相似的检测技术

基于视觉相似性的钓鱼网站检测，就是检验疑似钓鱼网页与目标合法网页的视觉相似度是否超过已设定的阈值来检测疑似钓鱼网页是否满足判断钓鱼属性的条件。基于视觉相似性的网络钓鱼检测技术以钓鱼网站的本质特征，即外观上与被仿冒的目标网站视觉性极为相似，作为判别标准则更具宽松性。

视觉检测技术大体分3个步骤：

##### 1）.. 疑似钓鱼网页代码转换成图片文件的格式。

##### 2）. 对转换的图片文件处理，得到相应的特征向量。

##### 3）. 由特征向量匹配目标网站的特征向量匹配。

德国的Max-Emanuel Maurer和Dennis Herzner提出一种基于视觉相似性[20, 21]的钓鱼检测方案，通过比较web页面中的图片之间的相似性和合法网站的图片可以检测到疑似钓鱼攻击。在服务器中存储了URL、屏幕截图等信息，其中索引文件包含带签名的截图。基于C/S结构，服务器把获取到的图像和库中预先签过名的合法网站图像在检测其中进行比对，并将结果返回给客户以提示。

加拿大的Scott Dick和James Miller提出了一种运用格式塔理论[22](Gestalt

Theory），将网页作为一个不可分割的实体，使用算法复杂性理论直接比较这些不可分割的页面，这可以有效的检测出钓鱼网页。格式塔理论为我们提供了我们的相似性识别方法的理论基础，其主要思想之一是整个感知形象不同于它各个隔离部分的总和。格式塔理论的基本公式可能以这种方式表达：存在整体的行为，这并不取决于个人的元素，但部分过程本身就是由本质决定的，这是格式塔理论的希望来确定这样的整体的性质。结构，网站的布局，和媒体设计创建了一个特定的感知，这比独立元素组成的Web页面更重要。

### 1.2.2 国内钓鱼网站检测发展及其现状

当前国内对于钓鱼网站的检测方法创新性不是很高，主要是针对网络钓鱼的攻击方式和防御方案进行技术手段的总结和定性研究和统计分析，钓鱼网站的统计和汇报则由中国反钓鱼网站联盟主导进行。中国反钓鱼网站联盟是国内

最为权威的监查汇报钓鱼网站的协调组织，目前拥有注册单位超五百多家。此联盟已建立及时解决的方案和策略，通过叫停非CN类域名和CN域名下的疑似钓鱼网站的解析等措施，达到减轻钓鱼网站的威胁和减少财产信息的损失和泄露。

哈尔滨工业大学工程硕士学位论文

目前就全世界的范围而言，钓鱼网站的现出率较高，已经对金融服务、网络商务的运营和维护产生了极大地影响，甚至有些是不可挽回的损失，在日常生活中则出现了很多网络盗号，账号密码盗取和篡改的现象。互联网在世界范围内是互通的，尤其在跨国贸易和访问方面，尤为突出的问题，即服务器的攻击和客户端的信息盗取已成为目前最为流行广泛的问题之一，通过侵入客户端的木马和恶意程序，而这些工具很大一部分则来源于不合法网站和恶意钓鱼网页的攻击。

根据最新关于钓鱼网站的处理简报所示，截至2015年4月，联盟累计认定并处理钓鱼网站230740个，2015年度钓鱼网站月处理情况[23]如图1-1所示。

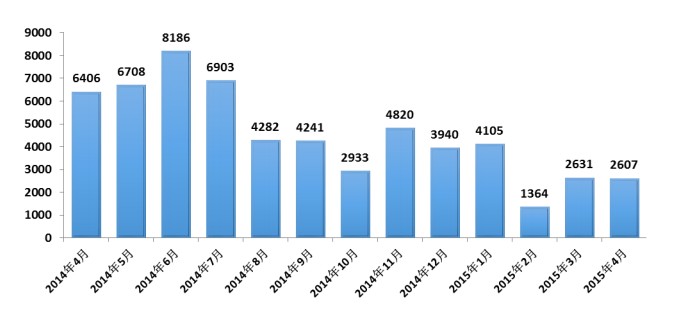


图1-1 2014-2015年度钓鱼网站月处理情况

相对较为乐观的是相较去年而言钓鱼网站的势头同比下降了不少，但依然存在着继续发展的情况。通过每月的简报分析所得，钓鱼网站的处理类型基本相似，在这里采用2015年4月份的统计结果作为代表数据。

4月份联盟接到的钓鱼网站举报中，波及到taobao网、中国工商银行ICBC、

银联、建设银行四家单位的钓鱼网站总量占全部举报量的绝大部分。其中高仿中国工商银行的钓鱼网站在检测的记录中一直居首，如图1-2所示。

哈尔滨工业大学工程硕士学位论文

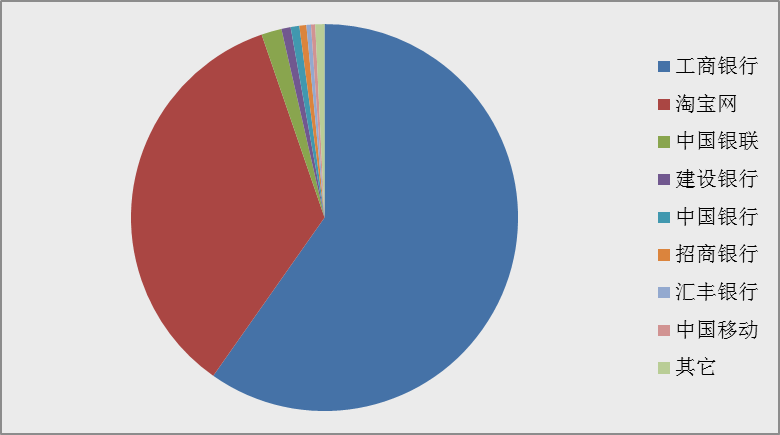


图1-2 钓鱼网站仿冒对象情况

钓鱼网站涉及到的行业包括金融和证券类、网上交易类、邮箱EMAL类，这三大类别。在这之中最高比例的当属金融类的钓鱼网站，其数量居于月处理总量的最大数额，占到了月处理总量的64.02%，如图1-3所示。

?

图1-3 钓鱼网站涉及行业分布图

该月被举报的CN域名的钓鱼的数目为193个，占本月处理的钓鱼网站总数的7.40%。由于不断细化CN域名的实名制工作和严格执行发放规则，使得

涉及CN域名的钓鱼网站数量的比例一直保持在较低的水平，如图1-4所示。

哈尔滨工业大学工程硕士学位论文

?

图1-4 CN钓鱼网站趋势图

使用. COM、. CC、. CN 和. TK 域名的钓鱼网站数量占本月处理总量的

87.53%。其中，. COM域名下的钓鱼网站占总钓鱼网站数量的比重高居首位。本月处理的钓鱼网站中，. GQ、. XYZ等非大众化域名下的钓鱼网站数量均有所增加，分别占到了月处理总量的3.30%和0.42%，如图1-5所示。

?

图1-5 钓鱼网站涉及顶级域名分布图

综上所述，目前钓鱼网站的发展现状严重影响网络应用的发展，极大地制约着网络发展的前进方向，人们在使用网络带给人们便利的同时自身的财产安全和个人信息的保密性受到很大冲击。由于互联网广泛的特点，钓鱼网站层出

不穷，深度模仿以假乱真，已经成为一个令人感到十分麻烦的问题。

哈尔滨工业大学工程硕士学位论文

因此，为了制约钓鱼网站的威胁和发展，我们需要创立一系列的检测方案和手段。这就需要在政府的大力支持下，各行各业不断完善网络安全的技术，及时发现及时上报，定期更新认证技术和推广新的应用。相信在业界共同努力下，钓鱼网站的威胁将变得越来越小。

## 1.3 本论文的主要工作内容

本课题研究的主要内容是部署在网络安全监测系统下的基于页面内容分析的钓鱼网站检测系统，其中完成业务需求下的概要设计和详细设计并完成系统中各个子模块的实现和接口的调用。该系统作为钓鱼网站的检测及汇报来源，构建在LINUX平台下，满足网络传输、数据统计、匹配检测等功能要求。

论文的组织结构主要分为以下几部分来介绍：

论文第1章从调研的角度介绍了课题的来源及课题的研究意义，以及与课题相关的国内外研究状况。

论文第2章从系统的需求入手，就功能需求和非功能需求的角度分别阐述了业务的要求，并分析了系统的业务需求。之后进行了相关技术原理的介绍和列举。就应用的角度说明各种技术在实现中的应用角度和应用方式。并给出相应的思路和解决方案。

论文第3章给出相关类图及流程图说明系统URL去重模块、白名单过滤模块、连接池模块、多模匹配模块、正则表达式匹配模块、日志模块的结构设计。

论文第4章描述了系统URL去重模块、白名单过滤模块、连接池模块、多模匹配模块、正则表达式匹配模块、日志模块的详细设计与实现，并通过流程图来描述各个模块内部的程序流程。

论文第5章按照设定的测试顺序分别对基于页面内容分析的钓鱼网站检测系统进行单元测试、集成测试，验证系统的功能和性能。

论文的最后内容是结论的描述，总结了该论文的调研成果和技术性方案的总结，并且给出了相关系统继续发展的阐述。

# 第2章系统需求与关键技术分析

哈尔滨工业大学工程硕士学位论文

本章主要内容主要分为以下几个部分：第一部分简要介绍系统的需求分析，涵盖了业务需求、功能需求和非功能需求在内的分析部分，介绍需要实现的几种主要业务模块是采用流程图的方式。第二部分描述了系统在开发的过程中需要采用并实现的关键技术，包括URL去重模块，另外由多模匹配算法和哈希算法等技术组成的技术解决思路，并给出相应的解决方案和措施。

## 2.1 需求分析

### 2.1.1 系统的功能需求

基于页面内容分析的钓鱼网站检测系统主要业务分为两大部分，预处理部分和检测部分。

预处理部分采用基于libevent的异步DNS解析和对URL的解析，拆分出IP列表、域名列表、路径等元素集合。在建立异步连接之后加入连接池，对连接池进行异步的I/O管理。对于异步的I/O管理则使用非阻塞的连接管理、超时管理和基于URL的大数据去重。其中对于数据的去重方面采用Bloom Filter，在大数据时代，有着很好的去重效果和较低的错误率。在对数据去重操作之后将会重构http数据包进行分发。

作为预处理部分最后一个部分，获取http数据包后进行解包，获得URL信息和其中内容。在不同的系统和编辑器下，编码的格式会有着截然不同的方式，考虑到系统兼容性，要对编码页面内容进行转码后才可以进行检测。此外还需对http状态码进行检测，对客户端和服务器的状态进行监测，使系统运行顺畅。另外，通过正则表达式匹配来抽取符合监测标准的特征信息，如：encoding，title，meta等。

检测部分进行页面内容的分词和分类。判断标题是否含有关键词，筛选出的关键词汇的过滤和匹配，进行特定检测条件下的分类。采用多模匹配算法可进行高效率的特征匹配。进行进一步的过滤和检测，结合whois 数据查询后的

合法性分析、基于哈希相似度的检测方法。

哈尔滨工业大学工程硕士学位论文

在以上各子模块的检测过程中，生成相应日志和URL检测报告。

基于页面内容的钓鱼网站检测系统可以分为两大部分，其中包含以下几部分模块：数据获取模块、内容解析模块、字符串匹配模块、日志模块等，其中系统架构如下图2-1所示。

?

图2-1 钓鱼网站检测系统架构图

URL的数据获取可通过几种模式来进行信息的采取，如读取文件中的URL

列表信息，和通过socket传入的相关URL信息。

页面内容解析模块从URL入手将目标地址中相关内容按照特定的检测抽取角度分解综合，形成分类的数据集并做下一步的检测和匹配。

多模匹配模块的重点在于构造一个类，将正则相关的一些操作封装在其中，在使用时，对于各部分正则匹配将会有选择的进行使用。

连接池部分即使用libcurl中的多线程中可操作的handle进行相关的操作，

Libcurl提供三种handle: easy\_handle、multi\_handle、share\_handl. easy\_handle: 为libcurl的最基础部分，所有的操作都是在easy\_handle上进行的，比如发送、请求数据都是在其上进行的。如果直接在easy\_handle 执行操作

curl\_easy\_perform 函数是阻塞的（即需要等到完成才返回）。

哈尔滨工业大学工程硕士学位论文

白名单过滤模块的设计和应用，在于通过配置文件选取特定类型的目标类进行筛选和检测，如检测银行欺诈类和博彩类的网站，则可以设置相应的匹配关键词来分类操作，对于解析之后的文本可以按照此类型进行判断分析分类。哈希相似度检验模块主要对于特征文本的匹配和检测则主要分关键词的哈

希匹配检测和图像的相关检测，设计中为保证测试的准确性和检出率的较严格要求，设置了两种检测模式，通过业务的需求和时间的实际情况酌情考虑设计并调试哪种方案。

钓鱼网站检测系统业务流程说明如下：

开启线程获得URL信息，在判断URL信息确定获取之后，初始化统计信息，获得待检测URL信息，无论是从文件中读入条目或者是通过网络连接传入系统，都需要进行一系列库存名单检测。通过黑白名单的筛选过滤，得到完全新增的陌生信息源输入，所需的信息源数据已经减少为需要相对较少工作量的数据了，这样才会进入下一步的业务流程。

所检测的URL信息为待检测疑似URL信息，但文件条目录入中的信息并非完全是不相同的，存在着重复上报的情况，由于新增URL数据信息的重复申请，存在着不同时段上报的数据存在着一定的重复性。这就需要去重业务的执行，来确保工作量的进一步减少和最终统计结果的无重复性。

下一步即为进行页面的DOM解析工作，从URL信息字符和HTML页面内容按标签解析两个角度进行关键词的匹配相关检测，和关于图像的一些检测手段的实施和执行。

从整个线程的执行过程来讲，主要分为以下几个步骤：

白名单过滤，首先就在相对减少冗余工作量的角度上减少不必要的检测；

URL 数据获取，将目标页面下载到本地进行检测前的数据准备工作；页面的

DOM树解析，解析出所需要的检测目标数据；关键词的匹配，对于检测到的可疑关键词进行按照一定方式的计算得分；哈希值的计算，通过一系列的权重配比计算可以通过设置各个得分层级的选择获得相应的分值，计算总得分时通过与设定的阈值来做比较判断出该特征关键词是否匹配疑似钓鱼网站。系统业务

功能流程如图2-2所示。

哈尔滨工业大学工程硕士学位论文

?

图2-2 系统总体功能流程图

基于以上业务流程，系统需要实现的主要业务大致可以分为三种：去重业务，特征文本匹配业务，哈希相似度检测业务。

哈尔滨工业大学工程硕士学位论文

去重业务流程如图2-3所示。

?

图2-3 去重业务流程图

去重的业务流程中，批量导入新增URL条目，会对每一条记录进行哈希算法的位存储，针对每一位进行哈希计算得到的结果进行位数组的置位，并填入1占位，以此类推，将每一条记录执行位存储标识的操作，并将此条信息记录在另一张列表之中进行记录。当此条目记录完成后进行下一条目的录入，完全依照上一条的存储原理进行，并进行按位查找，并判断此条目是否为已记录条目，如果是，则自动忽略此条目在新增筛选后列表中的信息，如果否，则记录在新

增条目中，依次类推直至完成原列表中所有条目均已完成检测操作业务。页面标签特征文本匹配主要业务流程如图2-4所示。

哈尔滨工业大学工程硕士学位论文



开始

否

是否匹配到title文本关键词

是

否

是否匹配到meta文本关键词

是

否

是否匹配到body文本关键词

是

结束

初始化变量

计算关键词得分

计算关键词得分

计算关键词得分

记录日志

图2-4 页面标签文本匹配业务流程

其中采用多模式匹配算法进行主要的执行流程，就是在很多连续不断的字符串中匹配到并寻找很多不同模式的特定的字符串。并通过相对较快的方法找出我们所需要的那种字符串，对于分词分类而言和关键字符串的匹配和过滤则有着很重要的应用。

通过多模匹配算法来进行每一项特征文本的匹配操作，针对各个标签部位

的特征文本匹配模型的导入，如“银行”，“密码”等标识文本的目的性检索，来进行目标字符的匹配检测，如果匹配到则记录到日志中待输出检查。

哈尔滨工业大学工程硕士学位论文

哈希检测业务主要作为文本相似度的匹配业务，业务流程如图2-5所示。



开始

否

是否大于阈值

是

结束

选取检测项

计算哈希距离

哈希值存入列表

判断距离值选取得分

依权重计算得分

对比阈值进行判定

为安全项

为疑似项

图2-5 文本相似度检测业务流程

通过哈希计算文本相似度的业务操作，可较好的完成钓鱼网站检测的任务，并有着较好的准确率。通过各个不同的web页面文本生成由哈希的方式所形成的指纹信息来当做校验的一种参考标准和手段。具体的算法比如最常用的MD5算法，这种设计的目的就是使得哈希计算之后的位置分布更加合理，更加分散，其中均匀的分布方式是我们所最希望看到的。针对着严谨细微的差别限制，输

入如果出现一点差异的话，哈希之后的结果都会有着极大的变化。可以看出，哈希计算之后的相似程度可以看做是直接反馈到输入信息的相似程度。

哈尔滨工业大学工程硕士学位论文

哈希计算的得分评价方式主要由计算哈希距离和加权计算求得分数，并通过设置一定的阈值来判定是否为文本相似。如果大于这个设定的阈值，会判定为疑似钓鱼网站，如果小于这个设定的阈值，则可以忽略为不相似项。

### 2.1.2 系统的非功能需求

系统要满足所有的非功能性需求包括：

##### 1）. 至少能够在周末的时间里稳定运行，产出日志文件结果正常。

##### 2）. 检测服务运行良好，保持系统稳定性。

##### 3）. 网络传输保持正常的访问速度和下载速率。

##### 4）. 能够通过简单的操作，方便部署到不同的设备。

### 2.1.3 系统预期达到的目标

系统的设计可以基本实现，能按照业务需求进行整合和运行，设备运行和系统的部署都需要配置完成并可以在系统出现故障时能方便的重新启动程序进行自启动。所有的模块和接口均可以执行自身的功能，运行良好。

在性能上，系统需要达到的目标是：

##### 1）. 系统部署的机器能够保证2天不间断的稳定工作，在周末无人的情况下也能保持正常的检测。

##### 2）. 系统检出率保持在稳定的水平，检测结果具有参考价值。

## 2.2 关键技术分析

基于页面内容分析的钓鱼网站检测系统的开发主要是部署在LINUX平台下。系统在开发和运行时的关键在于如何保持连接的畅通和有效地网络通信，在多线程之间的共同执行程序的运行，最终的目标是系统的稳定，URL去重，多模匹配算法的检测效果，连接池的稳定性和哈希相似度校验。以下给出几种在系统的设计和实现过程中需要用到的技术方案。

### 2.2.1 基于BloomFilter的URL去重处理

哈尔滨工业大学工程硕士学位论文

预处理阶段，URL的去重部分基于Bloom Filter[24-27]的计算方式，作为一种可以充分利用内存空间的数据结构，BloomFilter可以说是效率很高而且科学合理。通过利用位数组来作为数据和监测的载体，并通过判断重重的元素的属性和数值来判断是否属于这个集合。

BloomFilter技术主要应用于URL去重的处理方面。

BloomFilter通过以容纳一定的错误率和误判率作为代价，换取高效的检测速率，由于对位元素的敏感，使得这种算法对于整体而言会出现相对较大的疏漏。有着很大的风险将不属于该集合的元素误认为是属于这个集合的。

所以在容错率较低的条件下，这种方案并不适合，由于此系统允许出现一定的误差和错误，而这种误差和错误并不影响最后的结果汇总。所以Bloom Filter得这种特性可以为我们的系统节省很大的存储空间。

在设计之中不可避免的会遇到内存方面和计算速率方面的选择，这的确很难，由于需求的不同导致着选择的不同，这就需要根据具体情况具体分析了。如果需要保持一种较高的速率的话，对内存的需求可以说是很高的，自然对设备和开发的难度就要提升一个档次。如果内存占用率不可超过某一界限，则会对效率产生一个极大的抵消。

然而我们在BloomFilter[28-29]这种技术方面看到的是另一种处理方式，就是引入了错误率这个关键因素。由于大部分系统不会容忍将错误保持在正常的检测项目中，故尔这是很特别的处理方式。

我们可以认为这样的错误率出现的原因是在于在检测中，不属于特定集合的元素会被误认做属于该集合中的一员。增加了错误率的引入后，我们就可以节省很大的内存空间了。

在具体计算中初始状态时用于保存哈希计算结果的位数组的每一位的初始数据都被事先置位为0，作为初始化已经写入。对于含有n个元素的一个集，我们会使用k组不同的，相互独立的哈希函数，通过计算得出相应的映射对象，

并列入到映射的范围之中。每一个元素可以被置到不同的位上，置位的位置所对应的位数组元素，则有0置位为1。如下图2-6中所示。

哈尔滨工业大学工程硕士学位论文

说明: http://p.blog.csdn.net/images/p_blog_csdn_net/jiaomeng/275417/o_bf2.jpg

图2-6 哈希置位示意图 1

在判断这个集中的y1和y2是否包含在内，通过加载k次哈希函数到y上，如果所有经过计算后的位置上的元素的值都是1，我们则可以认为y命中，否则就认为y不命中。如下图2-7所示。

说明: http://p.blog.csdn.net/images/p_blog_csdn_net/jiaomeng/275417/o_bf3.jpg

图2-7 哈希置位示意图 2

对于其中的置位和哈希计算如上图所示，置位的方案也明显的可以看到，0位和1位被均匀的置位，哈希计算最不应该产生数据分布不均的问题。

### 2.2.2 多线程同步技术

本钓鱼网站检测系统采用多线程同步的工作方式，共享数据的存在使得我们不得不考虑这个问题，数据结构的通用和共用使得我们必须能进行共同的修改和变动。为了避免变量的访问冲突，我们就必须解决这个问题，否则更会造成内存泄露等更加严重的不可知情况。

我们需要进行对线程的同步和协调[30]的操作方案，来解决保持系统的稳定性这样的技术难题。当前的处理方案主要是利用互斥锁，临界区，信号量等措施，本系统中选取互斥锁来解决线程间的同步。

数据资源的状态，会被标识为是否能被访问，主要是靠信号灯[31]来判断区分，而标明是否能被访问则需要互斥锁来标注。操作线程调用内存中的资源就是用到互斥锁的技术。

互斥锁[32-33]的目的就在于控制系统运行期间的访问中的互斥现象，对于互斥锁而言，一但开启互斥锁，同一时刻仅能存在同一个任务线程在执行，一旦

任务执行完毕，要想解锁这个互斥锁，只能由这个任务本身自己解锁，其它任务线程不会影响到它的执行。

哈尔滨工业大学工程硕士学位论文

一但先执行的任务进行到尾声时，在其后申请的任务则会被阻塞，直到第一组任务完成时，解锁，第二个任务才会被执行到。直到自身释放互斥锁，当释放完毕时才会真正地结束。

调用基本的解锁，上锁操作可进行信号的操作，在由上锁到解锁的期间，变量是无法进行更改的，这也更好的保证了数据在进程间的同步。

### 2.2.3 多模匹配算法

多模匹配算法[34-36]在这里指的是在一个字符串中寻找多个模式字符字串的问题。给出已知的一个长字符串，和很多待匹配的短字符串，我们需要在其中找出匹配的字符串。多模匹配算法应用于关键字过滤、分词分类等问题中。多模匹配问题[37-40]的解决方法一般有Trie树算法，AC算法和WM算法等。

##### 1）hash算法，如图2-8所示。

?

图2-8 hash算法示例

优点：有效而且相对较为简单。缺点：需求空间较大。

##### 2）Trie树：

建立跳转路径避免回朔如图2-9所示。

哈尔滨工业大学工程硕士学位论文

?

图2-9 Trie树示例

跳转路径建立的算法思想：

如果要建立节点n1大于n2跳转路径需要满足：

1）n1和n2有相同值，代表同一个字符

2）n1的深度大与n2的深度

##### 3）对于n1节点的父节点F1，和n2节点的父节点有F1-> F2

优点：查询效率较高

缺点：会浪费一定的空间，数据结构较为复杂。

### 2.2.4 哈希相似度检测算法

哈希计算文本相似度的业务操作，可较好的完成钓鱼网站检测的任务，并有着较好的准确率。通过各个不同的web页面文本生成由哈希的方式所形成的指纹信息[41]来当做校验的一种参考标准和手段。具体的算法比如最常用的MD5算法，这种设计的目的就是使得哈希计算之后的位置分布更加合理，更加分散，其中均匀的分布方式是我们所最希望看到的。针对着严谨细微的差别限制，输入如果出现一点差异的话，哈希之后的结果都会有着极大的变化。可以看出，哈希计算之后的相似程度可以看做是直接反馈到输入信息的相似程度。

哈希计算的得分评价方式主要由计算哈希距离和加权计算求得分数，并通

过设置一定的阈值来判定是否为文本相似。如果大于这个设定的阈值，会判定为疑似钓鱼网站，如果小于这个设定的阈值，则可以忽略为不相似项。

哈尔滨工业大学工程硕士学位论文

Simhash[42-45]是由Moses Charikar提出的局部敏感哈希的一种。使用传统hash可能会产生如下的结果：

引用

Irb(main):006:0> p1 = 'the cat sat on the mat' irb(main):005:0> p2 = 'the cat sat on a mat' irb(main):007:0> p3 = 'we all scream for ice cream' irb(main):007:0> p1. hash

=> 415542861

Irb(main):007:0> p2. hash

=> 668720516

Irb(main):007:0> p3. hash

=> 767429688

使用simhash会应该产生类似如下的结果：引用

Irb(main):003:0> p1. simhash

=> 851459198

00110010110000000011110001111110

Irb(main):004:0> p2. simhash

=> 847263864

00110010100000000011100001111000

Irb(main):002:0> p3. simhash

=> 984968088

00111010101101010110101110011000

海明距离的定义，为两个二进制串中不同位的数量。上述三个文本的

simhash结果，其两两之间的海明距离为(p1, p2) =4, (p1, p3) =16以及(p2, p3) =12.

事实上，这正好符合文本之间的相似度，p1和p2间的相似度要远大于与p3的。如何实现这种hash算法呢？以上述三个文本为例，整个过程可以分为以下

哈尔滨工业大学工程硕士学位论文

六步：

##### 1） 选择simhash的位数，请综合考虑存储成本以及数据集的大小，比如说

32 位

##### 2） 把simhash的各个位的值置位初始化为0。

3)取原文本中的特征参数。比如对于“I am a little boy"，采用分词的方式得到如下结果集：{" i"," am"," a "," li"," tt"," le"," b"," oy"}

4)使用传统的32位hash函数计算各个word的hashcode，比如：“th". hash

= -502157718, “he". hash = -369049682, ……

5）对各部分的字符串匹配到的哈希码的每一位，如果该位为1，则哈希到的相应位的值加1；否则减 1

6）对最后得到的32位的simhash，如果该位大于1，则设为1；否则设为0。

## 2.3 本章小结

本章首先从钓鱼网站的检测角度入手说明基于页面内容分析的钓鱼网站检测系统的业务流程，介绍了各个主要业务模块的流程和功能需求。

从功能和需求方面入手，由整体到部分介绍了业务的大致流程，最后就系统按照需求所需要的技术方案的角度，介绍了当前为了实现检测算法所选用的技术手段。并以此为依据，逐步向实现靠拢，为完成整个系统的功能实现做技术支持。

# 第3章系统分析与设计

哈尔滨工业大学工程硕士学位论文

## 3.1 系统总体设计

### 3.1.1 系统框架设计

基于页面内容的钓鱼网站检测系统可以分为两大部分，其中包含以下几部分模块：数据获取模块、解析模块、数据匹配模块、日志模块等。系统层次结构图如图3-1所示。

?

图3-1 系统层次结构图

### 3.1.2 系统流程设计

哈尔滨工业大学工程硕士学位论文

系统业务功能流程如下图3-2所示。

?

图3-2 系统总体功能流程图

从整个线程的执行过程来讲，主要分为以下几个步骤：

哈尔滨工业大学工程硕士学位论文

白名单过滤，首先就在相对减少冗余工作量的角度上减少不必要的检测；

URL 数据获取，将目标页面下载到本地进行检测前的数据准备工作；页面的

DOM树解析，解析出所需要的检测目标数据；关键词的匹配，对于检测到的可疑关键词进行按照一定方式的计算得分；哈希值的计算，通过一系列的权重配比计算可以通过设置各个得分层级的选择获得相应的分值，计算总得分时通过与设定的阈值来做比较判断出该特征关键词是否匹配疑似钓鱼网站。

### 3.1.3 系统模块设计

该检测系统需要实现8个接口，分别是导入配置文件接口、初始化全局变量接口、线程参数引入/开启接口、白名单匹配接口、HTML页面下载接口、DOM解析接口、关键字段匹配接口、计算哈希值与总得分接口。

钓鱼网站检测流程中的主业务流程以各种接口调用来实现具体的功能，通过一系列初始的配置之后，在对线程的管理中通过开启新线程来进行各个步骤的实现。

业务功能结构图如下，如图3-3所示。

初始化设置

导入配置文件

初始化全局变量

线程参数导入/开启

Thread中进行

黑白名单匹配

HTML页面内容下载

DOM树解析

关键信息匹配

计算哈希值

计算哈希总得分

线程后管理

线程超时管理

配置文本保存模式

本地保存检测结果

图3-3 系统业务功能结构图

在整个流程中位于初始化设置中的函数接口描述为导入配置文件、初始化全局变量、线程参数导入/开启；位于线程之中的接口描述为黑白名单匹配、HTML 页面内容下载、DOM 树解析、关键信息匹配、计算哈希值、计算哈希总得分；位于线程结束后的后续工作中的功能接口主要由线程超时管理、本地保存检测结果、配置文本保存模式。

哈尔滨工业大学工程硕士学位论文

下面简单介绍这些业务处理接口的基本功能：

1. 导入配置文件：入口函数和配置文件举例为app\_run(". /config/app\_phishi

ng\_detect. config“）； 其中导入了检测所需要的参数和数据信息。

2. 初始化全局变量：由配置文件路径所引入的参数. /config/app\_phishing\_de tect. config作为函数接口init\_global\_var(config\_path)；中的实参，来实现对全局变量的初始化设置。

3. 线程参数引入/开启：创建线程来进入正式的检测流程接口实例为pthrea d\_create(&worker, NULL, app\_engine\_thread, & i);

4. 白名单匹配：暂定实现白名单的匹配，由于钓鱼网站一般会较频繁的更

换URL信息，制作和维护黑名单的成本开销较大，故用白名单作为筛选标准。白名单添加函数接口实例为g\_lock\_var(Lock\_white\_list)；为保证顺利完整的导入白名单到列表中，在其中增加了线程锁，在进行上锁和解锁的过程中保证了传入的完整性和可靠性。

5. HTML页面下载：通过URL下载页面内容，入口函数实例为url\_downl oader(url, html, log\_info, threadId)，导入URL信息和线程ID，并保存HTML页面和相关日志的保存。

6. DOM树解析：通过此函数gumbo\_parse(html. c\_str（）)进行DOM树的建立，并通过提取Title和Meta标签属性来获得标签对应的字段，并为进行关键词的相关匹配做准备。

7. 关键字段匹配：通过此接口来实现，bool find\_keywords(string text, uint 16\_t output\_type, uint32\_t &output\_param, string &log\_info, int32\_t thread\_id);对于第一个参数来讲，是区分title和meta等关键属性的选择参数，并做已自动筛选并存入日志和输出表列中。

8. 计算哈希值：通过计算文本的哈希值的函数实例cal\_hash\_by\_df(feature\_

哈尔滨工业大学工程硕士学位论文

text）；来获得选出特征的哈希值，并计算文本的逻辑距离cal\_him\_distance(hash\_i nfo. hash\_val, g\_simhash\_list[i]. hash\_val)；并加以判断，如果在一定的范围内则认为是相似的，否则就不是。

9. 计算总得分：总得分即将keyword的得分和simhash的得分加权配比的到相应的总分数进行阈值的判断。

10. 线程超时管理：通过设定clock来解决，clock. start（）；和clock. stop（）；函数实现计时性的判断。

11. 配置文本保存模式和本地列表的保存：函数接口实现为conf\_write(". /re sult. xml"," UTF-8", g\_res\_xml\_list);配置保存格式为UTF-8，存文件为XML文件。

## 3.2 系统功能模块设计

### 3.2.1 URL去重模块设计

去重模块采用基于Bloomfilter算法的字符串去重操作进行，Bloom-Filter一般用于在大数据量的集合中判定某元素是否存在。例如邮件服务器中的垃圾邮件过滤器。

URL的去重部分基于Bloom Filter的计算方式，作为一种可以充分利用内

存空间的数据结构，BloomFilter可以说是效率很高而且科学合理。通过利用位数组来作为数据和监测的载体，并通过判断重重的元素的属性和数值来判断是否属于这个集合。

BloomFilter通过以容纳一定的错误率和误判率作为代价，换取高效的检测速率，由于对位元素的敏感，使得这种算法对于整体而言会出现相对较大的疏漏。有着很大的风险将不属于该集合的元素误认为是属于这个集合的。所以在容错率较低的条件下，这种方案并不适合，由于此系统允许出现一定的误差和错误，而这种误差和错误并不影响最后的结果汇总。所以Bloom Filter得这种特性可以为我们的系统节省很大的存储空间。

在设计之中不可避免的会遇到内存方面和计算速率方面的选择，这的确很

难，由于需求的不同导致着选择的不同，这就需要根据具体情况具体分析了。如果需要保持一种较高的速率的话，对内存的需求可以说是很高的，自然对设备和开发的难度就要提升一个档次。如果内存占用率不可超过某一界限，则会对效率产生一个极大的抵消。然而我们在BloomFilter这种技术方面看到的是另一种处理方式，就是引入了错误率这个关键因素。由于大部分系统不会容忍将错误保持在正常的检测项目中，故而这是很特别的处理方式。

哈尔滨工业大学工程硕士学位论文

我们可以认为这样的错误率出现的原因是在于在检测中，不属于特定集合的元素会被误认做属于该集合中的一员。增加了错误率的引入后，我们就可以节省很大的内存空间了。

在具体计算中初始状态时用于保存哈希计算结果的位数组的每一位的初始数据都被事先置位为0，作为初始化已经写入。对于含有n个元素的一个集，我们会使用k组不同的，相互独立的哈希函数，通过计算得出相应的映射对象，并列入到映射的范围之中。每一个元素可以被置到不同的位上，置位的位置所对应的位数组元素，则有0置位为1。

在判断这个集中的y1和y2是否包含在内，通过加载k次哈希函数到y上，如果所有经过计算后的位置上的元素的值都是1，我们则可以认为y命中，否则就认为y不命中。

对于其中的置位和哈希计算如上图所示，置位的方案也明显的可以看到，0位和1位被均匀的置位，哈希计算最不应该产生数据分布不均的问题。

因而本系统中此模块需事先设计2组哈希算法模型：算法模型1：

While(\*key) {

H^=(h<<5) +(h>>2) +(unsigned char) \*key++

}

算法模型2：While(\*key) {

H^=(h<<6) +(h<<16) -h+(unsigned char) \*key++

}

哈尔滨工业大学工程硕士学位论文

此模块的设计依上面算法所述，可知bloomfilter是一个封装了哈希函数指针和包特定长度的位数组，对位数组的相关操作，比如添加和查找，清除等操作也一并封装在一起。

设计BLOOM类如下图3-4所示。

|  |
| --- |
| <<struct>>BLOOM |
| +asize : int  +a : unsigned char  +nfuncs : int |
| +bloom\_create() : BLOOM  +bloom\_destroy() : int  +bloom\_add() : int  +bloom\_check() : int  +\*hashfunc\_t() : unsigned int |

图3-4 BLOOM类图

此类为了保存计算字符串的hash算法之后的各种输出，对于此后的查询和清除等操作提供一个动态的存储的空间。下面给出bollmfilter模块含操作部分的业务流程图。如图3-5所示。

哈尔滨工业大学工程硕士学位论文

|  |
| --- |
| 填入生成新的列表 |



开始

是否匹配到已记录字串

否

是

跳至下一条记录

结束

初始化变量并传入参数

运行两组哈希函数并创建

bloomfilter

导入列表进行补位

导入列表进行过滤

进行去重操作

图3-5 bloomfilter去重模块业务流程图

模块的开始会进行相应的初始化配置，初始化结构体和参数，之后进行bloomfilter的创建，即选择至少两组哈希函数作为计算的依据，通过第一组列表每条记录的读入，来进行哈希计算并填充相应的位。

当待检测的记录录入时，通过再次计算哈希函数得到的位进行查找是否匹配到所述的文本哈希位记录，如果找到相同的则调制下一条记录进行继续查找，否则将填入新的列表之中，次列表作为去重结果作为下一模块的素材。

### 3.2.2 白名单过滤模块设计

哈尔滨工业大学工程硕士学位论文

基于URL检测是指通过检测网页的URL，就能区别当前所访问的网站是否是钓鱼网站。当前反钓鱼网站浏览器工具栏插件是最受欢迎的产品，可以广泛部署解决方案打击网络钓鱼网站，大部分的工具栏使用着黑白名单作为检测的数据库名单，方便维护和管理。

通过查看黑名单/白名单过滤(B/W)数据库确定URL。检查结果将被交付用户与警告用户可能的钓鱼网站，或通报网站是合法的。

尽管URL通过黑白名单匹配的检测技术实现容易而且命中率较为可观，但是却很难发现不在名单中的钓鱼网站是否是合法的，且确认黑名单需要大量动用人工力量去验证，效果经费投入比很低。

通过简单的匹配、对比实现对网页的检测来分析结果，黑名单的获取和确定需要人工进行审核和校对，具有一定时间上的滞后性，而钓鱼网站的存活期相对较短，而且变化很快，因此这类方法并未在实时性方面上解决网络钓鱼攻击带来的问题。

针对网络新增URL信息，往往含有正规网站的更新URL的出现，面对大量新增数据，需要经过黑白名单过滤来减少一些工作量。

目前暂定不使用黑名单的原因在于，经调研和分析得来的结论，非法仿冒网站的URL信息生存时间特别短，数量特别多，加入黑名单的话会增加过滤的成本和时间，效果也不见得会高很多，意义不大。

而白名单则可以把确定正规的网站排除掉，这样剩下的无论是确定是钓鱼网站还是疑似钓鱼网站都可以列入检测名单中，这样可以避免漏查。

白名单过滤部分白名单的处理和存储类图如图3-6所示。

哈尔滨工业大学工程硕士学位论文

|  |
| --- |
| <<struct>> compare\_str |
| +p1 : char  +p2 : char |
| +operate() : bool |

|  |
| --- |
| <<struct>> basic\_string |
| +string : char |
|  |

|  |
| --- |
| <<type>> str\_int32\_map |
| +string : char  +int32\_t : int |
| +str\_hash() : int  +compare\_str() : bool |

图3-6 白名单过滤部分白名单存储部分类图

其中白名单需要通过hashmap进行处理，通过pushback操作将处理过后的白名单中的每一条记录添加进去，并与用vector维护的URL进行比对筛选。

新增域名白名单过滤主要是读取白名单中的合法域名，并将URL列表中相应内容清除并生成新的URL列表。白名单过滤业务流程图如图3-7所示。

哈尔滨工业大学工程硕士学位论文



开始

否

是否获取成功

是

开启互斥锁

读取白名单信息

否

是否获取成功

是

筛选结果放到集合中

关闭互斥锁

写入日志

结束

域名变量初始化

获取新增URL列表

记录错误日志

图3-7 白名单过滤业务流程图

白名单过滤模块主要以匹配关键字筛选合法域名信息为手段，将处理后的新增域名信息保存在集合中，待处理之后的操作进行各种检测。

针对线程锁的关闭也同样采用相同的处理，即执行g\_unlock\_var(uint16\_t l ock\_type)的操作和选项，期间将相应信息记录日志。接口domain\_filter(string u rl, string &log\_info, int32\_t thread\_id)，指定了在特定线程之下的操作URL信息的过滤，并将相关信息记录到日志中。

接口的实现从初始化开始，通过判断获取URL信息列表queue <string> g\_u rl\_list是否成功来决定是否进行下一步，如果读取信息成功，则开启线程锁，进行筛选和保存的过程，在结束之后关闭线程锁。其中对于开启锁的选项有两个，g\_lock\_var(uint16\_t lock\_type)中通过switch语句选择开启的是URL列表读入或是白名单列表，由pthread\_mutex\_lock(& g\_mutex\_url\_list)，和pthread\_mutex\_lo

哈尔滨工业大学工程硕士学位论文

ck(&g\_mutex\_white\_list)分别进行开启。

### 3.2.3 连接池模块设计

连接池部分是解决异步域名解析和下载HTML 页面内容的前提，可基于

libcurl实现。

连接池部分即使用libcurl中的多线程中可操作的handle进行相关的操作，

Libcurl提供三种handle: easy\_handle、multi\_handle、share\_handl. easy\_handle: 为libcurl的最基础部分，所有的操作都是在easy\_handle上进行的，比如发送、请求数据都是在其上进行的。如果直接在easy\_handle执行操作curl\_easy\_perform 函数是阻塞的（即需要等到完成才返回）。

multi\_handle: libcurl为异步操作提供的接口，允许调用方在一个线程中处理多个操作（就是easy\_handle上的操作，注意是单线程下的），内部multi\_handle采用堆栈的方式保存多个easy\_handle，然后在一个线程中可以同时对多个

easy\_handle进行处理，multi\_handle的执行操作curl\_multi\_perform函数是立即返回的，不会阻塞。

另外libcurl还可以支持断点续传的功能，即自动记录传输的文件的位置，便于续传时能够避免重复操作所带来的问题，值得关注的问题就是，一个线程每次只能使用一个handle。

下列为连接池模块体系设计类图，如图3-8所示。

哈尔滨工业大学工程硕士学位论文

|  |
| --- |
| <<struct>> compare\_str |
| +p1 : char  +p2 : char |
| +operate() : bool |

|  |
| --- |
| <<struct>> basic\_string |
| +string : char |
|  |

|  |
| --- |
| <<type>> str\_int32\_map |
| +string : char  +int32\_t : int |
| +str\_hash() : int \*  +compare\_str() : bool |

|  |
| --- |
| <<struct>> ac\_automata\_t |
| +root : node  +current\_node : node  +all\_nodes\_num : unsigned int  +all\_nodes\_max : unsigned int  +total\_strings : unsigned long  +all\_nodes : node |
| +ac\_automata\_create() : ac\_automata\_t  +ac\_automata\_reset() : void |

\*

图3-8 连接池部分设计类图

|  |
| --- |
| <<struct>> node |
| +outgoing  +matched\_id : int  +matched\_count : short  +matched\_max : short  +outgoing\_degree : unsigned short  +outgoing\_max : unsigned short  +depth : unsigned short  +final : short |
| + node\_create() : node  +node\_next\_find() : node  +node\_next\_update() : node  +node\_next\_index\_find() : node  +node\_findbs\_next() : node |

|  |
| --- |
| <<struct>>edge |
| +alpha : char  +alpha\_count : unsigned short  +next : node |
| +node\_outgoing\_register() : void  +node\_matched\_register() : void  +node\_sort\_edges() : void  +node\_free() : void |

接下来执行curl\_easy\_perfrom将会链接到远程host去执行待发命令，实时接收将要返回的数据。在确定准确地接收到数据时，将要所需要饿的数据读到本地缓冲区中来。

页面内容解析的首要问题就在于html页面的完整下载，通过基于libcurl的html页面下载完成了主体html页面的代码获取。数据资源的状态，会被标识为是否能被访问，主要是靠信号灯[31]来判断区分，而标明是否能被访问则需要互斥锁来标注。操作线程调用内存中的资源就是用到互斥锁的技术。

互斥锁[32-33]的目的就在于控制系统运行期间的访问中的互斥现象，对于互斥锁而言，一但开启互斥锁，同一时刻仅能存在同一个任务线程在执行，一旦任务执行完毕，要想解锁这个互斥锁，只能由这个任务本身自己解锁，其它任务线程不会影响到它的执行。一但先执行的任务进行到尾声时，在其后申请的任务则会被阻塞，直到第一组任务完成时，解锁，第二个任务才会被执行到。

直到自身释放互斥锁，当释放完毕时才会真正地结束。

哈尔滨工业大学工程硕士学位论文

调用其打开关lock，开开关unlock的操作方法即可进行信号的操作，在由上锁到解锁的期间，变量是无法进行更改的，这也更好的保证了数据在进程间的同步。通过采取多线程同步的技术可以不断的建立连接，而不影响正在检测的线程，这是很可取的。

### 3.2.4 页面内容解析模块设计

页面内容解析模块的功能主要是通过访问目标URL，并对相应的HTML页面代码进行下载，并通过相应的统一转码转换成可统一处理的可读字符串对象进行进一步的处理，并记录错误日志和相应的解析文档。

在下载完成后，接下来介绍DOM树解析部分的设计，这是基于gumbo 的

HTML页面解析的模块，使用gumbo\_parse或者gumbo\_parse\_with\_options就可以得到一个GumboOutput数据结构，就可以从该结构中寻找想要的东西了。

先给出DOM解析部分的设计类图，如图3-9所示：

\*

|  |
| --- |
| <<struct>> GumboOutput |
| +root : GumboNode  +document : GumboNode  +errors : GumboVector |
| +GumboOutput() : GumboOutput  +gumbo\_parse\_with\_options() : GumboOutput  +gumbo\_parse\_with\_options() : void |

|  |
| --- |
| <<struct>> GumboVector |
| +data : void  +length : unsigned int  +capacity : unsigned int |
|  |

|  |
| --- |
| <<struct>> GumboInternalOptions |
| +userdata : void  +tab\_stop : int  +stop\_on\_first\_error : bool  +max\_errors : int |
|  |

|  |
| --- |
| <<struct>>GumboNode |
| +index\_within\_parent : int |
|  |

|  |
| --- |
| RegexTool |
|  |
| +analyzeCharSet() : bool  +stripHTML() : char  +stripSymmetricTags() : char  +getMatchedStr() : bool  +isMatched() : bool  +getNumOfMatchs() : int |

|  |
| --- |
| HtmlProcessor |
| -regex\_tool : RegexTool |
| +cleantext() : char  +get\_body\_text() : char  +get\_title\_and\_meta() : bool  +get\_feature\_text() : char  -get\_text\_tags\_num() : unsigned int |

图3-9 DOM解析设计部分类图

通过向目标服务器发送请求，可通过传输协议下载目标HTML页面，页面内容下载流程原理图如下所示，如图3-10所示。

哈尔滨工业大学工程硕士学位论文

?

图3-10 页面下载流程原理图

通过gumbo结构部分处理HTML页面并进行DOM解析，处理标签属性所对应的文本的下载。再由HtmlProcesor类进行下载后的提取等操作，如文本特征的选项及导入，获取title和meta文本的函数。

哈尔滨工业大学工程硕士学位论文

### 3.2.5 多模匹配模块的设计

本系统采用基于ac自动机的多模匹配算法，AC算法本质上来说和Trie树一样，ac自动机的构造如下：

##### 1). 构造一支Trie树，作为ACautomachine的检索数据结构。

2）. 构造fail指针，即为了继续配取当前字符落单时，自动跳转到具有最长前后缀公共的字符处。ACautomachine在匹配时如果遇到匹配失败目标字符时，则利用fail指针来进行跳转。

3）. 扫描主串进行匹配。

下面简介一下基于ac多模匹配算法的设计部署：

建立Trie的过程：

1）. 指针p，指向当前匹配的字符。若p指向root，表示当前匹配的字符序列为空。（root是Trie入口，没有实际含义）。

2）. 指针p-> fail, p的失败指针，指向与字符p相同的结点，若没有，则指向root，在建立fail指针时有寻找与p字符匹配结点的作用，在扫描时作用最大

##### 3）. 指针temp，测试指针

对于Trie树中的一个节点，对应一个序列s[1... m]。此时，p指向字符s[m]。若在下一个字符处失配，即p-> next[s[m+1]] == NULL，则由失配指针跳到另一个节点（p-> fail）处，该节点对应的序列为s[i... m]。若继续失配，则序列依次跳转直到序列为空或出现匹配。在此过程中，p的值一直在变化，但是p对应节点的字符没有发生变化。在此过程中，我们观察可知，最终求得序列s则为最长公共后缀。另外，由于这个序列是从root开始到某一节点，则说明这个序列有可能是某些序列的前缀。

构造失败指针：

用BFS来构造失败指针，与KMP算法相似的思想。root入队，第1次循环

时处理与root相连的字符，也就是各个单词的第一个字符h和s；第2次进入循环后，队列弹出h，接下来p-> h节点的fail指针，并指向root节点；p=p-> fail也就是p=NULL说明匹配序列为空，则把节点e的fail指针指向root表示没有匹配序列然后节点e进入队列；第3次循环时，弹出的节点a，接下来p-> a节点的fail指针，并指向root节点，随后将a指向root的fail指针并入队；第4次进入循环时，弹出节点h由于指向下一节点的指针指向的内容为空，便把h节点的fail指针，指向右边那个root的子节点h，以此类推。

哈尔滨工业大学工程硕士学位论文

扫描：

构造好Trie和失败指针后，我们就可以对主串进行扫描了。这个过程和KMP算法很类似，但是也有一定的区别，主要是因为AC自动机处理的是多串模式，需要防止遗漏某个单词，所以引入temp指针。

匹配过程分两种情况：

##### 1)字符匹配，表示沿树的一直可以到达目标字符，只需沿该路径遍历向下一个节点继续执行匹配操作。

##### 2）字符不匹配，则返回到上一节点处继续匹配，匹配过程结束的标注即为指针指向root即根节点。

重复这2个过程中的任意一个，直到模式串走到结尾为止。

其中多模匹配部分操作的设计类图如下图3-11所示。

哈尔滨工业大学工程硕士学位论文

|  |
| --- |
| <<struct>> ac\_automata\_t |
| +root : node  +current\_node : node  +all\_nodes\_num : unsigned int  +all\_nodes\_max : unsigned int  +total\_strings : unsigned long  +all\_nodes : node |
| +ac\_automata\_create() : ac\_automata\_t  +ac\_automata\_reset() : void  +ac\_automata\_locate\_failure() : void  +ac\_automata\_search() : int  +ac\_automata\_find() : int  +ac\_automata\_free() : void |

图3-11 文本匹配部分类图

|  |
| --- |
| <<struct>> node |
| +outgoing  +matched\_id : int  +matched\_count : short  +matched\_max : short  +outgoing\_degree : unsigned short  +outgoing\_max : unsigned short |
| +depth : unsigned short  +final : short |
| + node\_create() : node  +node\_next\_find() : node  +node\_next\_update() : node  +node\_next\_index\_find() : node  +node\_findbs\_next() : node |

|  |
| --- |
| <<struct>> int\_set\_t |
| +max : unsigned int  +item : unsigned char  +num\_el : unsigned int  +first : unsigned int |
| +int\_set\_create() : int\_set\_t  +int\_set\_init() : void  +int\_set\_head() : unsigned int  +int\_set\_suc() : unsigned int  +int\_set\_count() : unsigned int  +int\_set\_empty() : int  +int\_set\_insert() : int  +int\_set\_remove() : void  +int\_set\_search() : int  +int\_set\_free() : void |

|  |
| --- |
| <<struct>>edge |
| +alpha : char  +alpha\_count : unsigned short  +next : node |
| +node\_outgoing\_register() : void  +node\_matched\_register() : void  +node\_sort\_edges() : void  +node\_free() : void |

由图可知这些操作是与关键词列表所比对类进行匹配的，通过vector迭代器来进行对g\_keywordList的顺序查找。

### 3.2.6 正则表达式匹配模块设计

正则匹配模块主要对检测内容进行特定的筛选，在页面代码全部下载之后，对于页面层次结构进行特定选取，获得关键标签内容的字符串，主要在host，

port, filename等方面的关键信息，还有html代码中可以明显描述的meta，title，

body等部分代码，正则匹配是找寻关键内容检测的主要步骤之一，是所有匹配检测的必要过程。经由匹配之后的关键词和关键信息则会保存在集合vector中，方便提取和检测。

正则匹配类中为主要的项目操作，有以下几部分组成，如表3-1所示。

表3-1 正则匹配项目表

哈尔滨工业大学工程硕士学位论文

| 函数名 | 匹配对象 |
| --- | --- |
| analyzeCharSet | 分析字符集 |
| extractAllPictsstatic | 解析图片 |
| extractTitle | 匹配 title 文本 |
| extractMeta | 匹配 meta 文本 |
| getUrl\_host | url 中匹配主机信息 |
| getUrl\_port | url 中匹配端口信息 |
| getUrl\_fileName | url 中匹配文件信息 |
| RegexMatchhtml | 匹配 html |

还有正则配置项：static bool getConfig\_attribute(string content, string &key, string &value);作为正则匹配之前的准备工作。

### 3.2.7 哈希相似度检测模块设计

网页页面内容匹配计算阶段是本模块的最直接阶段。本阶段的主要目标之一，就是对于在URL正则匹配阶段中，不确定为正规网站，但是依然未能做出判定的疑似网站。通过对网页内容进行分词、特征项提取形成待检内容，随后系统将待检测字符集与仿冒目标的相应位置的文本库进行匹配比对，判断出待检对象是否为命中目标。

哈希匹配检测模块的设计思路在于，根据所选取的特征来计算所选取的比对目标进行哈希值的计算，并将按照一定规则计算出的哈希值进行存储，形成一个哈希值的列表。在对相应对象进行哈希值计算之后还会对此列表进行哈希距离的计算和进一步的比对，并存储相应的信息。计算哈希值得分之后还会存在一项值的判定，根据不同的取值阶段界定相关特征的归属，并进行进一步的分类。

其中在进行整体的哈希值计算得分的时候，存在特定的判定和计算公式，在本案例设计中得分按照score = keyword\_score\*0.3 + simhash\_score\*0.7；这样的计算方法来判定，相应的按照一定的流程计算出特征值和哈希值的总体得分，并依权来计算相应的重要特征是否更加有效。在计算出相应的总得分之后，根

据设定的阈值来判定是否为疑似钓鱼行为。

哈尔滨工业大学工程硕士学位论文

哈希检测模块主要由以下几个接口调用来实现哈希值的计算和比较。如下图所示，simhash模块相关类图如图3-12所示。

|  |
| --- |
| <<struct>>SimHashInfo |
| +hash\_val : unsigned long  +name : char |
|  |

|  |
| --- |
| SimHasher |
|  |
| +cal\_hash\_by\_df() : unsigned long  +cal\_him\_distance() : unsigned int  +cal\_hash\_patch() : int  +save\_hash\_val() : void  +load\_hash\_file() : void  +handl\_sim\_hash\_line() : bool |

|  |
| --- |
| <<struct>> url\_s |
| +url : char  +host : char  +ip : char  +path : char  +html : char  +logo : char  +whois : char  +image : char |
|  |

图3-12 simhash模块类图

|  |
| --- |
| <<struct>> urlpair\_t |
| +\*purl : url\_s  +surl : url\_s  +sim\_file : char  +probility : |
| double |

接口static uint32\_t cal\_him\_distance(uint64\_t lhs, uint64\_t rhs)；用于实现计算哈希距离。

接口static uint32\_t cal\_hash\_patch(string dir\_path, string store\_file, vector

<SimHashInfo> \*hash\_list)；则用来计算匹配结果。

接口static void save\_hash\_val(SimHashInfo hash\_info, string file\_path);用于保存哈希值。

用于保存哈希值的日志文件接口static void load\_hash\_file(string file\_path, vector<SimHashInfo> & hash\_list)；由此来引入特定的计算功能。并在整体的流程中解决相应的问题和实现特定的功能。

哈希计算的得分评价方式主要由计算哈希距离和加权计算求得分数，并通过设置一定的阈值来判定是否为文本相似。如果大于这个设定的阈值，会判定为疑似钓鱼网站，如果小于这个设定的阈值，则可以忽略为不相似项。

### 3.2.8 日志模块设计

哈尔滨工业大学工程硕士学位论文

日志模块是为了解决在周末人员不在单位，而且又需要每天准时发报告和日志的情况下而设计的。日志模块记录的是各个模块在工作时的状态信息和监控信息输出，为维护系统和及时发现错误日志信息提供帮助。

日志模块的功能就是记录系统在不同阶段模块的运行状态信息的关注点的备份和输出。日志模块运行在各个重要的位置上。先对初始化日志记录文件，操作文件配置，写入相关的配置并创建对应格式的日志。

在整个系统运行的过程中，在缓冲区中放置，不断将设定的内容及格式通过写文件的形式写入日志。日志模块的工作流程图如图3-13所示。



开始

初始化

否

初始化是否成功

是

否

是否有日志数据

是

将数据存入缓存

缓存中是否有 否

数据

是

写入文件

缓存是否为空

否

是

工作是否结束

否

是

结束

图3-13 日志模块工作流程图

日志模块运行和部署在系统的各个模块之中，在执行过程中不断输出相关信息并形成文本记录到本地。

哈尔滨工业大学工程硕士学位论文

## 3.3 本章小结

本章根据系统的功能需求进行设计，论述了系统的主要功能处理模块的功能。并详细介绍了URL去重模块、白名单过滤模块、连接池模块、多模匹配模块、正则表达式匹配模块、日志模块的设计。给出各个功能模块的图表，为系统的最终实现做好了实现的准备。

# 第4章系统的实现

哈尔滨工业大学工程硕士学位论文

## 4.1 URL去重模块实现

此模块的实现依上面算法所述，介绍一下实现的描述。可知bloomfilter是一个封装了哈希函数指针和包特定长度的位数组，对位数组的相关操作，比如添加和查找，清除等操作也一并封装在一起。

在初始化的时候需要确定位数组的长度，综合文献上的资料可设定范围为相对可靠的长度，可由伪码所示：

设定位数组长度：

#define SETBIT(a, n)

(A[n/CHAR\_BIT] | = (1 < <(n%CHAR\_BIT))；

设定位的计算方式：

#define GETBIT(a, n)

(A[n/CHAR\_BIT] | = (1 < <(n%CHAR\_BIT))；

读取位的计算方式：

Bloom-> a=

Calloc((size+CHAR\_BIT-1) /CHAR\_BIT, sizeof(char))；

其实二者的计算过程是一样的，只不过就是先录入的字符串计算后填入的位先记录并置位，之后的只需查找当前位是否被置位过。

此类为了保存计算字符串的hash算法之后的各种输出，对于此后的查询和清除等操作提供一个动态的存储的空间。

可知在bloomfilter的实现中我们需要准备几部分的设计

1）哈希函数的设计，准备几项哈希函数并规范化命名。

2）设计置位和取位的函数。

3）准备两个文件进行检测的条目读取和检测结果的输出。

下面给出bollmfilter模块含操作部分的程序流程图。如图4-1所示。

哈尔滨工业大学工程硕士学位论文



开始

初始化变量并开辟内存空间

从文件中获取URL条目字符串

|  |  |
| --- | --- |
| 建立哈希JSHASH,RSHASH... | |
|  |  |

|  |  |
| --- | --- |
| 导入列表进行补位 | |
|  |  |

|  |  |
| --- | --- |
| 位数组置位操作setbit | |
|  |  |



否

是否有被置位getbit

是

否

检测条目是否检测完

是

结束

检测结果保存在日志中

跳至下一哈希进行检测

图4-1 bloomfilter去重模块流程图

模块的开始会进行相应的初始化配置，初始化结构体和参数，之后进行bloomfilter的创建，即选择至少两组哈希函数作为计算的依据，如unsigned int RSHash(unsigned int len, char\* str);和unsigned int JSHash(unsigned int len, char\*

str）；其中字符串作为输入，通过相应的哈希计算我们可得到一个整形的哈希值。

通过第一组列表每条记录的读入，来进行哈希计算并填充相应的位。

哈尔滨工业大学工程硕士学位论文

当待检测的记录录入时，通过再次计算哈希函数得到的位进行查找是否匹配到所述的文本哈希位记录，如果找到相同的则调制下一条记录进行继续查找，否则将填入新的列表之中，次列表作为去重结果作为下一模块的素材。

## 4.2 白名单过滤模块实现

白名单过滤模块主要以匹配关键字筛选合法域名信息为手段，将处理后的新增域名信息保存在集合中，待处理之后的操作进行各种检测。

接口domain\_filter(string url, string &log\_info, int32\_t thread\_id)，指定了在特定线程之下的操作URL信息的过滤，并将相关信息记录到日志中。

接口的实现从初始化开始，通过判断获取URL信息列表queue <string> g\_u rl\_list是否成功来决定是否进行下一步，如果读取信息成功，则开启线程锁，进行筛选和保存的过程，在结束之后关闭线程锁。其中对于开启锁的选项有两个，g\_lock\_var(uint16\_t lock\_type)中通过switch语句选择开启的是URL列表读入或是白名单列表，由pthread\_mutex\_lock(& g\_mutex\_url\_list)，和pthread\_mutex\_lo

ck(&g\_mutex\_white\_list)分别进行开启。

**g\_lock\_var(Lock\_white\_list);**

**if( (brand.length()>0 && g\_white\_list.find(brand) != g\_white\_list.end()) || (rootDo main.length()>0 && g\_white\_list.find(rootDomain)!=g\_white\_list.end()) ||g\_white\_list.fin d(domain) != g\_white\_list.end())**

**{**

**log\_info += "1\n"; g\_unlock\_var(Lock\_white\_list);**

**return true;**

**}**

**g\_unlock\_var(Lock\_white\_list); log\_info += "0\n";**

导入的白名单则存入构造过的hash\_map中，通过互斥锁结构体pthread\_mu tex\_t进行控制，控制代码部分逻辑如下：

针对线程锁的关闭也同样采用相同的处理，即执行g\_unlock\_var(uint16\_t l

ock\_type）的操作和选项，期间将相应信息记录日志。

哈尔滨工业大学工程硕士学位论文

其中白名单需要通过hashmap进行处理，通过pushback操作将处理过后的白名单中的每一条记录添加进去，并与用vector维护的URL进行比对筛选。

## 4.3 连接池模块实现

当模块入口程序接口调用时，需要准备好的URL信息和相应的文件读写操作准备。先需要初始化libcurl 库，libcurl 的初始化只需进行一次，执行

curl\_easy\_init（）完成操作，libcurl中使用到被称为easy interface的api函数，所有这些函数都是有相同的前缀：curl\_easy，使用easy interface之前需要创建一个easy handle句柄，这个句柄就会用于执行每次操作。

在其中的线程均有属于自己的，用于数据通信的句柄，线程之间不可共享

Easy handle。

除了两种特殊情况之外的情况，就是信号以及SSL/TLS句柄，在这两者之外的libcurl都是线程安全的。对于加密通信信道传输是所需要的访问方式，用于多线程的都应该满足库的一些操作条件。

接下来执行curl\_easy\_perfrom，连接到远程host，将要下载的数据读入到本地缓冲区。

页面内容解析的首要问题就在于html页面的完整下载，通过基于libcurl 的

html页面下载完成了主体html页面的代码获取。连接池模块的流程图4-2如下所示。

哈尔滨工业大学工程硕士学位论文

?

图4-2 连接池模块程序流程图

依据页面下载的原理可知，需通过目标URL获取目标的ip，这就需要在这里首先进行DNS解析的操作。DNS解析的目的就是在于把我们所称作域名的字符串翻译成对应的IP地址，然后去访问该IP地址，从而获得相应的网页。

连接池部分即使用libcurl中的多线程中可操作的handle进行相关的操作，

Libcurl提供三种handle: easy\_handle、multi\_handle、share\_handl. easy\_handle: 为libcurl的最基础部分，所有的操作都是在easy\_handle上进行的，比如发送、请求数据都是在其上进行的。如果直接在easy\_handle 执行操作

curl\_easy\_perform 函数是阻塞的（即需要等到完成才返回）。

哈尔滨工业大学工程硕士学位论文

multi\_handle: libcurl为异步操作提供的接口，允许调用方在一个线程中处理多个操作（就是easy\_handle上的操作，注意是单线程下的），内部multi\_handle采用堆栈的方式保存多个easy\_handle，然后在一个线程中可以同时对多个

easy\_handle进行处理，multi\_handle的执行操作curl\_multi\_perform函数是立即返回的，不会阻塞。

下面介绍异步DNS解析部分程序流程图，如图4-3所示：

?

图4-3 异步DNS解析部分程序流程图

另外libcurl还可以支持断点续传的功能，即自动记录传输的文件的位置，便于续传时能够避免重复操作所带来的问题。

哈尔滨工业大学工程硕士学位论文

Html页面解析模块接口：bool getURLDataBycurl(const char\* URL, string

& content, int & contentLen)；通过传入char\* URL指针传入URL信息，指定初始化curl句柄curlInit(CURL \*& curl, const char\* url)将URL绑定在句柄中，在之后的设置和执行过程中调用句柄去完成相应的工作。

通过以上说明所呈现出来的大致流程可用以下时序图来表，如下图4-4 所

示。



curlInit(CURL\*curl,chonst char\*)

curl\_easy\_setopt(CURL\*curl,const char\*,char\*buf)

bool curl\_easy\_perform(CURL \*curl);

writer(void \*data, int size, int nmemb, void \*param)

code

curl\_easy\_getinfo(CURL \*curl); code

curl\_easy\_cleanup(CURL\*curl)

getURLDataBycurl

curl

writerParam

图4-4 连接池部分时序图

## 4.4 页面内容解析模块实现

页面内容解析由页面内容下载和页面内容解析两部分组成。

其中Urlvisiter类包含了全部的主流程的操作和数据属性的声明，在这些属性的定义中，则较多的使用了结构体而不单单采用变量的形式作为保存的媒介，

而其中包含的这些操作所直接相关联的操作类则有PosixRegexer所维护，涉及到具体的下载目标的区分和相关操作。在UrlVisiter中的主要操作包含了初始化，统一转码，项目选择，文本的读写操作。

哈尔滨工业大学工程硕士学位论文

针对下载HTML页面的中的特定检测内容，在PosixRegexer类中主要设计了以下几部分的操作，现列入检测目标操作表中作为区分，如表4-1所示。

表4-1 页面内容关键属性检测表

| 检测项目 | 操作 |
| --- | --- |
| Title 文本导出 | ExtractTitle(string html, string &title); |
| Meta 文本导出 | ExtractMeta(string html, size\_t htmlLen, string &meta); |
| 图片导出 | ExtractAllPicts(string html, size\_t htmlLen, vector<string> &urls); |
| Host 信息导出 | GetUrl\_host(string url, string &host, string &rootDomain, string &brand); |
| Port 信息导出 | GetUrl\_port(string url, string &port); |
| File 信息导出 | GetUrl\_fileName(string url, string &fileName); |

由此表可见，通过以上操作，可以将所需要的信息各自下载下来进行暂存以待进一步的DOM解析。

示例代码：

Int main(int argc, char\*\* argv) {

GumboOutput\* output = gumbo\_parse(argv[1]);

// Do stuff with output-> root gumbo\_destroy\_output(&kGumboDefaultOptions, output);

}

在下载完成后，接下来介绍DOM树解析部分的实现，这是基于gumbo 的

HTML页面解析的模块，使用gumbo\_parse或者gumbo\_parse\_with\_options就可以得到一个GumboOutput数据结构，就可以从该结构中寻找想要的东西了。

通过gumbo结构部分处理HTML页面并进行DOM解析，处理标签属性所对应的文本的下载。再由HtmlProcesor类进行下载后的提取等操作，如文本特征的选项及导入，获取title和meta文本的函数。

解析出来的特定标签属性的文本保存在变量中，继续进行下一阶段的关键词匹配模块。

## 4.5 多模匹配模块实现

哈尔滨工业大学工程硕士学位论文

在本系统中，多模匹配算法具体采用AC算法，即有限自动机匹配。

ac自动机的构造如下：

1). 构造一支Trie树，作为ACautomachine的检索数据结构。

2）. 构造fail指针，即为了继续配取当前字符落单时，自动goto跳转到具有最长前后缀公共的字符处。ACautomachine在匹配时如果遇到匹配失败目标字符时，则利用fail指针来进行跳转。

3）. 扫描主串进行匹配。

其中goto函数是一个状态在接受一个字符后转向另一个状态或者失败的函数，构造goto的伪代码如图4-5所示。

?

图4-5 构造goto的伪码图

下面简介一下建立Trie的过程：

哈尔滨工业大学工程硕士学位论文

1）. 指针p，指向当前匹配的字符。若p指向root，表示当前匹配的字符序列为空。（root是Trie入口，没有实际含义）。

2）. 指针p-> fail, p的失败指针，指向与字符p相同的结点，若没有，则指向

root，在建立fail指针时有寻找与p字符匹配结点的作用，在扫描时作用最大

3）. 指针temp，测试指针对于Trie树中的一个节点，对应一个序列s[1... m]。此时，p指向字符s[m]。若在下一个字符处失配，即p-> next[s[m+1]] == NULL，则由失配指针跳到另一个节点（p-> fail）处，该节点对应的序列为s[i... m]。若继续失配，则序列依次跳转直到序列为空或出现匹配。

failure函数伪代码如图4-6所示。

http://static.oschina.net/uploads/space/2014/0127/194613_nt7E_227203.jpg

图4-6 构造failure函数的伪码图

在此过程中，p的值一直在变化，但是p对应节点的字符没有发生变化。在此过程中，我们观察可知，最终求得序列s 则为最长公共后缀。另外，由于这

个序列是从root开始到某一节点，则说明这个序列有可能是某些序列的前缀。关键字的集合作为入口的输入项，输出项则为结果的状态集。

哈尔滨工业大学工程硕士学位论文

构造失败指针：用BFS来构造失败指针，与KMP算法相似的思想。

root入队，第1次循环时处理与root相连的字符，也就是各个单词的第一个字符h和s；

第2次进入循环后，队列弹出h，接下来p-> h节点的fail指针，并指向root节点；p=p-> fail也就是p=NULL说明匹配序列为空，则把节点e的fail指针指向root表示没有匹配序列然后节点e进入队列；

第3次循环时，弹出的节点a，接下来p->a节点的fail指针，并指向root节点，随后将a指向root的fail指针并入队；第4次进入循环时，弹出节点h由于指向下一节点的指针指向的内容为空，便把h节点的fail指针，指向右边那个root的子节点h，以此类推。

## 4.6 正则表达式匹配模块实现

正则匹配的第一部分是对URL做正则，主要对host，port，filename等有关部分的关键部分进行匹配。

对于host信息的接口表示为static int getUrl\_host(string url, string &host, string &rootDomain, string & brand)；

对于端口的相关检测则可以用此接口static bool getUrl\_port(string url, string & port); static bool getUrl\_scheme(string url, string & scheme)；

而对于url中涉及的文件名称则可以采用此接口来进行匹配static bool getUrl\_fileName(string url, string & fileName) 。

对于html页面标签内容的筛取则主要由title, meta等标签内所对应的信息来检测，如提取title主要内容的接口bool PosixRegexer:: extractTitle(string html, string & title)。

而对于meta则可以表示为bool PosixRegexer:: extractMeta(string html, size\_t htmlLen, string & meta) 。

在接口static int getUrl\_host(string url, string &host, string &rootDomain,

string & brand）的实现中，包括ip、域名、根域名的匹配筛选，分别匹配之后集合来保存结果。

哈尔滨工业大学工程硕士学位论文

其中也包含img图片url提取相关的接口bool PosixRegexer:: extractAllPicts (string html, size\_t htmlLen, vector <string> & urls)，将图片信息提取出来并保存。

下面列出提取标签文本的正则文本和对应属性，如表4-2所示。

表4-2 标签文本正则表示

| 属性 | 正则匹配操作 |
| --- | --- |
| Title | <title>((.|\\r|\\n)\*?)</?title> |
| Meta | <meta[^>]\*?charset=\"?'?([^'\" >]\*) |
| Imag | <img[^>]\*?src=\"?'?([^\"> ]+) |
| Domain | (^|://)(([a-z0-9]([-a-z0-9]\*[a-z0-9])?\\.)+ (com|net|edu|biz|gov|org|in(t|fo)|([a-z][a-z])))($|[/?:# ]|\\r|\\n) |
| Root | (([^./]+)((\\.(com|net|edu|biz|gov|org|in(t|fo))\\.  ([a-z][a-z]))|(\\.(com|net|edu|biz|gov|org|in(t|fo)))|(\\.([a-z][a-z]))))($|[/?:# ]|\\r|\\n) |
| Port | :([0-9]{1,5})($|[/?]) |
| Scheme | ^(mailto|ssh|ftp|https?):// |
| Ip | (^|[/])(([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\\.  ([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\\.  ([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\\.  ([01]?[0-9][0-9]?|2[0-4][0-9]|25[0-5]))($|[/?:# ]|\\r|\\n) |

正则匹配类中为主要的项目操作，有以下几部分组成：

Static bool RegexMatch(const std::string &\_pattern, const std::string &\_input, std::vector <std::string> \*\_output, std::string \*\_err\_msg);

html页面标签项和图片项：

static bool analyzeCharSet(string html, string &charSet);

static bool extractAllPicts(string html, size\_t htmlLen, vector<string> &urls);

static bool extractTitle(string html, string &title);

哈尔滨工业大学工程硕士学位论文

static bool extractMeta(string html, size\_t htmlLen, string &meta);

URL解析项的正则匹配：

Static int getUrl\_host(string url, string &host, string &rootDomain, string &brand);

Static bool getUrl\_port(string url, string & port);

Static bool getUrl\_scheme(string url, string & scheme);

Static bool getUrl\_fileName(string url, string & fileName);

还有正则配置项：static bool getConfig\_attribute(string content, string &key, string &value);

依据正则原理中所述，针对疑似URL为入口参数进行正则匹配，可以通过

一系列规则来提取出相应字段和符号间的关键文本，并进行关键词的匹配。

## 4.7 哈希相似度检测模块实现

根据所选取的特征来计算所选取的比对目标进行哈希值的计算，并将按照一定规则计算出的哈希值进行存储，形成一个哈希值的列表。在对相应对象进行哈希值计算之后还会对此列表进行哈希距离的计算和进一步的比对，并存储相应的信息。计算哈希值得分之后还会存在一项值的判定，根据不同的取值阶段界定相关特征的归属，并进行进一步的分类。

通过以下处理将计算特征文本的哈希汉明距离。

Hash\_info. hash\_val = SimHasher:: *cal\_hash\_by\_df*(feature\_text);

其中根据汉明距离判断得到的哈希值如下代码所示。

**if**(min\_dist <= 3)

simhash\_score = 1;

**else if**(min\_dist <= 7) simhash\_score = 0.9;

**else if**(min\_dist <= 16) simhash\_score = 0.8;

**else if**(min\_dist <= 24) simhash\_score = 0.6;

**else**

哈尔滨工业大学工程硕士学位论文

Simhash\_score = 0.4;

其中在进行整体的哈希值计算得分的时候，存在特定的判定和计算公式，在本案例设计中得分按照“得分=关键词得分\*0.3 + sim哈希得分\*0.7”这样的计算方法来判定，相应的按照一定的流程计算出特征值和哈希值的总体得分，配对计算可能性得分如：“可能性配对得分= 0.6 + (得分- 0.6) /(0.75-0.6) \*0.4”此类计算方案可由相关系统的设定来进行设计，代码如下。

Score = keyword\_score\*0.3 + simhash\_score\*0.7;

其中相似度的计算得分的代码如下所示。

**If**(score> = 0.6)

{

Log\_info +=" yes"; pair\_t\_p. result = 1;

**Strncpy**(pair\_t\_p. sim\_file, min\_hash\_info. name. c\_str(), 255); pair\_t\_p. probility = 0.6 + (score - 0.6) /(0.75-0.6) \*0.4; pair\_t\_p. probility = pair\_t\_p. probility> 1? 1: pair\_t\_p. probility;

}

**else**{

Log\_info +=" no"; pair\_t\_p. probility = score;

}

最后根据设定的阈值来判定是否为钓鱼疑似项，阈值设定为0.6，大于此值的特征项为存在钓鱼嫌疑项，小于此值的为安全项。

## 4.8 日志模块实现

日志模块的实现主要是在线程锁的控制下进行文件的读和写，在部署到各模块的程序之中运行，即当每部分模块检测流程结束之后，会在相应的日志信息中增添一些条目，记录我们所关注的信息点和故障信息。

日志模块是必不可少的，在程序和系统的评价方面，日志的参考价值是很大的，可知检测结果和方便查取异常。

日志模块的功能就是记录系统在不同阶段模块的运行状态信息的关注点的备份和输出。日志模块运行在各个重要的位置上。先对初始化日志记录文件，操作文件配置，写入相关的配置并创建对应格式的日志。

哈尔滨工业大学工程硕士学位论文

日志模块一直运行在各个模块之中直到程序被退出，程序流程如下图4-7

所示。

?

图4-7 日志模块程序流程图

初始化各项参数，在确定初始化完毕的情况下，加互斥锁并打开文件进行数据流的写入，当该检测部分完成的时候关闭文件，并删除内存中的数据，之后释放互斥锁，并终止日志记录的线程。

在整个系统运行的过程中，在缓冲区中放置，不断将设定的内容及格式通

过写文件的形式写入日志。

哈尔滨工业大学工程硕士学位论文

## 4.9 本章小结

本章根据系统设计的内容对各部分模块的实现进行了详细的描述。

钓鱼网站检测流程中的业务流程以各种接口调用来实现具体的功能，通过一系列初始的配置之后，在对线程的管理中通过开启新线程来进行各个步骤的实现。

在整个流程中位于初始化设置中的函数接口描述为导入配置文件、初始化全局变量、线程参数导入/开启。

位于线程之中的接口描述为白名单匹配、HTML页面内容下载、DOM树解析、关键信息匹配、计算哈希值、计算哈希总得分；位于线程结束后的后续工作中的功能接口主要由线程超时管理、本地保存检测结果、配置文本保存模式。

# 第5章系统的测试和分析

哈尔滨工业大学工程硕士学位论文

本章主要介绍对系统进行测试的测试条件，以及测试环境的搭建，并在测试中设定相应规范，测试方案可以设定为单元测试、系统测试和集成测试，下面就这三个测试形式进行介绍。

## 5.1 测试环境

系统部署在服务器上，测试的运行环境如表5-1所示。

表5-1 测试环境详细表

| 名称 | 配置 |
| --- | --- |
| 服务器 | Linux 系统：版本 CentOS-6.5-i386 |

## 5.2 单元测试

单元测试针对系统中各模块单元，以这些模块单元为对象进行测试。在对程序内部结构解析和代码测试为主的白盒测试，以及对各模块以日志为反馈的黑盒测试之下设置测试用例和测试角度的选取，并检查是否有错误出现。

对于完整性的测试和检验而言，检查内部数据在单元测试的过程中检验能否完整的执行。对于边界的数据测试，主要检测承载程度和校验水准。也要检查是否满足特定的逻辑覆盖测试。

在综合以上的测试前提下，主要进行了接口测试和局部的数据结构的测试。

### 5.2.1 接口测试

接口测试主要是判断的流入和流出模块的验证数据是否正常可靠，是否衔接顺畅，在模块调用时参数部分的执行元素是否匹配。检查接口不一致的现象，并清查找不到函数所属正确的入口的情况等进行了主要的检查。

主控程序模块主要使用的接口有app\_run，init\_global\_var，pthread\_create，g\_lock\_var, gumbo\_parse, find\_keywords，cal\_hash\_by\_df, conf\_write 以及

url\_downloader。

其接口测试的用例和结果如表5-2所示。

哈尔滨工业大学工程硕士学位论文

表5-2 主程序模块接口测试表

| 接口函数名 | 测试结果 |
| --- | --- |
| app\_run | 测试成功 |
| init\_global\_var | 测试成功 |
| pthread\_create | 测试成功 |
| g\_lock\_var | 测试成功 |
| gumbo\_parse | 测试成功 |
| find\_keywords | 测试成功 |
| cal\_hash\_by\_df | 测试成功 |
| conf\_write | 测试成功 |
| url\_downloader | 测试成功 |

综上，完成了对系统中各个功能子模块的单元测试，并确定了接口的功能都正常可用，参数设置和输入输出也是匹配的。

验证了系统中的数据在各功能模块中流入流出的正确性。

### 5.2.2 局部数据结构测试

为了数据的完全性和准确性，模块中临时的存储部分包含了这两种指标的测试项目。这就应用到了局部数据结构的测试，为了确保程序执行过程中的顺利。

针对局部数据的结构审查，主要采取代码审查的方式进行。制定了缺陷审查计划，和重点审查类型说明，来确定模块的数据结构是否相容或者是否合适。在适当的位置需要赋值来进行检测。

通过变量的属性来校验，从初始化方面和变量的默认值的角度，通过变量名的再校准，来检验数据结构中的溢出现象和各种异常。

对系统整体和局部细致的审查之后，确认局部数据结构的无异常，运行正确无误，确保在各模块的设计和实现中，数据缓存的正确性。综上，有了以上细致的测试之后可以保证在正常的情况下，系统运行是良好的。

## 5.3 集成测试

哈尔滨工业大学工程硕士学位论文

单元测试完毕之后需要进行下一步的测试，即为集成测试。当每个模块分别的都测试完毕，但不代表组合起来就不会出现各种各样的问题，故而需要进行集成测试。

将各部分模块按照需求组装起来形成一个完整性的检测步骤时，我们先进行功能测试。主要是对系统是否能按照预订的功能进行正确的工作表现，另外需要不断的检测各部分接口的问题是否会频繁出现。

集成程序做整体性测试，用作黑盒测试来检测，判断系统功能是否可以实现而且运行正常。

## 5.4 系统测试

在测试的最后一个阶段，将进行系统测试，并检验系统运行是否良好，并从日志产出和检测数据进行抽样检查，进行每日样本分析。

抽样测试结果如下所示，结果保存至result. txt文本中：

<root>

<iurl>

<name>null</name>

<gurl>

<gname>null</gname>

<surl>[www.sohu.com](http://www.sohu.com/)</surl>

<status>not phishing website:43.82%</status>

<whois>null</whois>

<html>null</html>

<image>null</image>

</gurl>

</iurl>

<surl>https://easyabc.95599.cn/SelfBank/netBank/zh\_CN/entrance/logonSelf.a

spx</surl>

哈尔滨工业大学工程硕士学位论文

<status>visit website failed</status>

<whois>null</whois>

<html>null</html>

<image>null</image>

</gurl>

</iurl>

</root>

抽样输出日志，保存在log. txt文本中，如下所示：

#url: [http: //www. ccb. com/cn/home/index. html](http://www.ccb.com/cn/home/index.html)

#domain\_filter:0

#url\_downloader:1

--convert\_charset:1

#title\_text:欢迎访问中国建设银行网站-个人客户

#meta\_text:金融，银行，建设银行，网上银行，电子银行，手机银行，股票行情，电话银行，外汇行情，黄金行情，基金，理财，商城，信用卡，金融资讯金融，银行，建设银行，网上银行，电子银行，手机银行，股票行情，电话银行，外汇行情，黄金行情，基金，理财，商城，信用卡，金融资讯

#title\_match--> #keyword\_matcher:1 建设银行

#meta\_match--> #keyword\_matcher:1建设银行，网上银行，手机银行，电子银行，信用卡，金融

#body\_match--> #keyword\_matcher:1建设银行, ccb. com,网上银行，网银，个人网银，账户，版权所有, ICP,服务热线，手机银行，电子银行，信用卡，证书，银行卡，公网安，建行，马上开通，功能演示,13030780,110102000450,金融

#keyword\_score:0.155039

#min\_simhash\_distance:0 19. htm

#simhash\_score:1

#total\_score:0.746512

哈尔滨工业大学工程硕士学位论文

#result: yes [http: //www. ccb. com/cn/home/index. html](http://www.ccb.com/cn/home/index.html)

\*\*--\*\*--\*\*--\*\*--\*\*--\*\*--\*\*--

#url: [http: //www. icbc. com. cn/icbc/](http://www.icbc.com.cn/icbc/)

#domain\_filter:0

#url\_downloader:1

--convert\_charset:1

#title\_text:

#meta\_text:

#title\_match--> #keyword\_matcher:0

#meta\_match--> #keyword\_matcher:0

#body\_match--> #keyword\_matcher:1工商银行，工行，网上银行，注册，账户，手机银行，e理财，电子银行，信用卡，存折，输入，金融

#keyword\_score:0.0676862

#min\_simhash\_distance:2 newicbc. htm

#simhash\_score:1

#total\_score:0.720306

#result: yes [http: //www. icbc. com. cn/icbc/](http://www.icbc.com.cn/icbc/)

\*\*--\*\*--\*\*--\*\*--\*\*--\*\*--\*\*--

由此可见系统运行良好，检测结果相对较为清晰详细，可以说明系统已经具备了需求下的基本实现，满足了业务和性能方面的需求。

## 5.5 本章小结

本章中，依照测试中的V模型对系统做了一系列测试工作，包含了单元测试，集成测试，系统测试三个部分。并对单元测试中的接口进行了测试，局部的数据测试。

在集成测试中，个部分模块不是单独的一部分了，做为一个整体，进行了完整的业务测试，验证了各部分接口的参数类型的准确定，参数个数的确定性，

以及接口的数据顺畅性。

哈尔滨工业大学工程硕士学位论文

在系统测试中，经验证，系统运行情况基本符合设计标准。通过长时间的运行测试和大批量的数据进行的压力测试表明，系统具备基本的可靠性和稳定性。

从测试结果日志可知，系统满足功能方面的需求以及非功能方面的需求的设定，性能指标方面也是达标的，可以应用到实际运行环境的部署中。

结论

哈尔滨工业大学工程硕士学位论文

论文首先通过对钓鱼网站检测的国内外现状进行阐述和研究，并且针对目前国家钓鱼网站检测的需求和钓鱼网站检测系统的实际要求进行分析，设计了基于页面内容分析的钓鱼网站检测系统。并对系统设计进行了需求分析和设计，并针对解决方案给出了相应的实现方法。

本论文的主要工作包含以下几部分：

1）设计了一个部署在linux服务器上的体系结构。整个系统主要的组成模块包括URL去重模块、白名单过滤模块、连接池模块、多模匹配模块、正则表达式匹配模块、日志模块等，更为清晰而且易于开发维护。

2）基于Bloomfilter的URL去重模块的解决了网页抓取部分针对URL重复抓取的情况，通过去重的处理减轻了工作量并提高了工作效率。白名单过滤模块的实现很好的进一步减轻了检测的工作量，通过匹配白名单模块去除合法性的URL条目，剩下的就是可以的待匹配检测的条目了。针对连接池模块，通过采用libcurl的技术和基于gumbo的技术很好的爬去网页内容并解析出我们想要检测的部分，这提供了清晰的检测解决方案，解析出title, meta等部分的关键词和body中的URL和图片信息以待进一步的检测。多模匹配模块先通过正则匹配，对URL做正则，主要对host，port，filename等有关部分的关键部分进行匹配，然后通过建立Trie树和fail指针较好的匹配出相关匹配字符，给出了较好的检测结果。哈希检测部分选取不同标准下的抽取来建立多维向量，通过检测得分的设计，评分按照权重进行加权计算最终得到检测结果。日志部分，保存了各标签部分匹配处的关键词，保留了哈希检测各部分的得分情况等关键检测信息。

3）通过多个层次的测试，验证了系统的稳定性和可靠性。通过可靠性测试，验证了系统部署后能支持120小时不间断的网络服务，能够满足周末非工作日仍能正常工作的要求。经过测试，系统的性能指标方面和满足了实际项目的要求，可靠性也满足需求，能够实现稳定的的检出和日志汇报生成等功能。

参考文献

哈尔滨工业大学工程硕士学位论文

[1] Anti-Phishing Working Group [EB/OL][. http: //www. antiphishing. org,](http://www.antiphishing.org/) 2008-0 1/2011-12-15.

[2] Microsoft Corporation. Microsoft phishing filter: a new approach to building trust in E-Commerce Content [R], White Paper 2008.

[3] Engin Kirda, Christopher Kruegel. Protecting Users against Phishing Attacks[J]. The Computer Journal, 2006, 49(05): 554-561.

[4] Christian Ludl, Sean McAllister, Engin Kirda, et al.. On the Effectiveness of Techniques to Detect Phishing Sites[C]. In Proc. of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Lucerne Switzerland, July 12-13 2007: 20-39.

[5] Steve Sheng, Brad Wardman, Gary Warner, et al.. An Empirical Analysis of Phishing Blacklists[C]. In Proc. of the sixth Conference on Email and Anti-Spam, California USA, July 16-17 2009.

[6] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, et al.. PhishNet: Predictive Blacklisting to Detect Phishing Attacks[C]. In Proc. of the IEEE INFOCOM, San Diego Canada, March 14-19 2010: 1-5.

[7] Ye Cao, Weili Han, Yueran Le. Anti-phishing Based on Automated Individual White-List[C]. In Proc. of the DIM'08, Virginia, USA, Oct. 31, 2008: 51-59.

[8] Sujata Garera, Niels Provos, Monica Chew, et al.. A Framework for Detection and Measurement of Phishing Attacks[C]. In Proc. of the WORM'07, Virginia USA, Nov. 2, 2007: 1-8.

[9] Colin Whittaker, Brian Ryner, Marria Nazif. Large-Scale Automatic Classification of Phishing Pages [C]. In Proc. of 17th Annual Network and Distributed System, California USA, Feb. 28-March 3 2010.

[10] Justin Ma, Lawrence K. Saul, Stefan Savage, et al. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs[C]. In Proc. of the KDD'09, Paris France, June 28-July 1, 2009: 1245-1254.

[11] Justin Ma, Lawrence K. Saul, Stefan Savage, et al. Identifying Suspicious URLs: an Application of Large-scale Online Learning[C]. InProc. of the 26th International Conference on Machine Learning, Montreal Canada, June, 14-18, 2009.

[12] Kurt Thomas, Chris Grier, Justin Ma, et al. Design and Evaluation of a Real-Time URL Spam Filtering Service[C]. In Proc. of the IEEE Symposium on Security and Privacy, California USA, May, 22-25, 2011.

[13] Aaron Blum, Brad Wardman, Thamar Solorio, et al. Lexical Feature based Phishing URL Detection using online Learning[C]. In: Proc. of the AISec'10, Chicago USA, Oct. 8 2010: 54-60.

哈尔滨工业大学工程硕士学位论文

[14] Steve Sheng. Brad Wardman. GaryWarner, et al. An Empirical Analysis of Phishing Blacklists[A]. In: Proc. of the 6th Conference on Email and Anti-Spam[C]. USA: ACM Press, 2009. 183-192.

[15] Sujata Garera, Niels Provos. Monica Chew, et al. A Framework for Detection and Measurement of Phishing Attacks[A], In: Proc. of the WORM'07[C]. USA: ACM Press, 2007.1-8.

[16] Justin Ma. Lawrence K. Saul. Stefan Savage, et al. Identifying Suspicious URLs: an Application of Large-scale Online Learning [A]. In: Proc. of the 26th InternationalConferenceonMachineLearning[C]. Canada: Madison, 2009.681-688.

[17] ZHANG, Y., HONG, J., CRANOR, L. 2007. CANTINA: A content-based approach to detecting phishingweb sites. InProceedings of the 16th International Conference on World Wide Web (WWW'07). 639–648.

[18] Salton, G. and M. J. McGill, Introduction to Modern Information Retrieval.

New York, NY: McGraw-Hill, 1986

[19] Phelps, T. A. and R. Wilensky, Robust Hyperlinks and Locations, D-Lib Magazinevol. 6(7/8), 2000. [http: //www. dlib. org/dlib/july00/wilensky/07wilensky.](http://www.dlib.org/dlib/july00/wilensky/07wilensky) html

[20] Max-Emanuel Maurer, Dennis Herzner. Using Visual Website Similarity for Phishing Detection and Reporting. CHI'12, May 5-10, 2012, Austin, Texas, USA. ACM 978-1-4503-1016-1/12/05.

[21] Anthony Y. Fu Liu Wenyin, Xiaotie Deng, et al.. Detecting Phishing Web Pages

With Visual Similarity Assessment Based on Earth Mover's Distance (EMD)[J]. IEEE Transaction on Dependable and Secure Computer, 2007, 3(04):301-311.

[22] Chen, T. -C., Dick, S., and Miller, J. 2010. Detecting visually similar Web pages: Application to phishing detection. ACM Trans. Intern. Tech. 10, 2, Article 5 (May 2010), 38 pages

[23] 中国反钓鱼网站联盟. 2015 年 4 月钓鱼网站处理简报 [R]. 2015.4.

[24] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. Internet Mathematics, 1(4): 485–509, 2005.

[25] M. Mitzenmacher. Compressed Bloom Filters. IEEE/ACM Transactions on Networking 10: 5 (2002), 604—612.

[26] 吴亮. 搜索引擎中网络爬虫的设计[J]. 决策与信息, 2008

[27] 王军, 彭建. 网络爬虫的结构设计研究[J]. 科技信息(学术版), 2007

[28] 王春梅. 基于BloomFilter的网络爬虫URL消重算法研究[J]. 产业与科技论坛, 2011(10): 18

[29]王岩. 搜索引擎中网络爬虫技术的发展[J]. 电信快报, 2008

哈尔滨工业大学工程硕士学位论文

[30] 王日宏. 基于VC 的Win32 多线程同步问题[J]. 计算机系统应用,

2004(7):60-62

[31] 穆兵. POSIX实时标准介绍[J]. 计算机工程与设计, 1992(3): 9-13, 74

[32] 赵慧斌, 李小群. Linux互斥锁机制的研究和改进[J]. 计算机科学, 2003(30): 8

[33] stallnigw. 操作系统内核与设计原理, 第四版[M]. 电子工业出版社, 2001.159~173

[34] Boyer R S, Moore J S. A fast string searching algorithm[J]. Communications of the ACM, 1977, 20(10): 762-772.

[35] Knuthd, Morris J, Prattv. Fast pattern matching instrings[J]. SIAM Journal on Computing, 1977, 6(1): 323-350.

[36] 戴英侠. 系统安全与入侵检测[M]. 北京: 清华大学出版社, 2002.

[37] 陈辰. 入侵检测系统的部署及入侵模式的识别[J]. 计算机系统应用, 2002(11): 104-106.

[38] 李璐, 王宏志, 李建中等. 一种优化的字符串相似连接算法[J]. 计算机研究与发展, 2009, 46(Suppl): 319-325.

[39] ARASU A, GANTI V, KAUSHIK R. Efficient exact set-similarity joins[C]. Proceedings of the 32nd International Conference on Very Large DataBases. [S. l.]: VLDB, 2006: 918-929.

[40] BAYARDO R J, MA Y, SRIKANT R. Scaling up all pairs similarity search[C]. Proceedings of the 16th International Conference on World Wide Web. NewYork: ACM, 2007: 131-140.

[41] 张祖平, 徐昕, 龙军等. 文本相似性度量中参数相关性与优化配置研究[J]. 小型微型计算机系统, 2011, 32(5): 983-988.

[42] Panagiotis Papadimitrion, Ali Dasdan, Hector Garcia-Molina. Webgraph sim ilarity for anomaly detection [J]. Journal of Internet Services and Applicat ions, 2010, 1(1): 19-30.

[43] Moses S Charikar. Similarity estimation techniques from rounding algorith ms[A]. In: Proceedings of 34th Annual ACM Symposium on Theory of Co mputing[C], ACM, 2002: 380-388.

[44] Gurmeet Singh Manku, Arvind Jain, Anish Das Sarma. Detecting near-duplicates for Web crawling[A]. In: Proceedings of the 16th International Conference on World Wide Web[C], ACM, 2007: 141-149.

[45] JiangQi-xia, Sun Mao-song. Semi-supervised SimHash for efficient document simlilarity search[C]. In: Proceedings of the 49th Annual Meeting of the Association for Computational Lingustics, 2011: 93-101.

哈尔滨工业大学学位论文原创性声明及使用授权说明

哈尔滨工业大学工程硕士学位论文

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于页面内容分析的钓鱼网站检测系统的设计与实现》是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：日期：年月日

学位论文使用授权说明

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：日期：年月日

导师签名：日期：年月日

致谢

哈尔滨工业大学工程硕士学位论文

两年的时光飞快，硕士阶段的学习和工作生活也到了末尾。在这期间我的导师李东教授给了我很大的帮助，老师严谨认真的治学态度、扎实的工作精神都让我受益匪浅，尤其是思维的指导使我深深的感到了教授的风采。

李东老师始终耐心的在论文的各个过程中为我解答疑惑，修改错误，并且很及时的为我审阅论文。特别感谢李东老师在很紧急的情况下依然会放弃一些自己非工作日中的时间来为我指导，回复邮件。对此，耽误了老师的休息时间深表歉意。

感谢我的校外导师葛瑞海，和我的项目负责人庹宇鹏老师。在论文的指导方面，葛老师在多年实践开发的基础上对我提出了很多有针对性的建议，指导我调研国外文献，让我感受到了科研工作者的严谨和视野。感谢他们在我生活上和工作上的帮助，不仅让我磨练了技术，也更加体会到无私奉献的精神。

感谢我的父母对我的培养和支持，让我有了奋斗的勇气。

感谢哈尔滨工业大学的各位老师和教授，感谢你们认真而耐心的指导和审核，百忙之中给与我宝贵的指导意见。

感谢北京赛思信安技术股份有限公司，为我提供了良好的工作环境。感谢我的朋友们在我撰写和修改论文时对我的帮助。

个人简历

哈尔滨工业大学工程硕士学位论文

** 教育历程：**

2009.09—2013.07 中国石油大学（华东） 电气工程及其自动化

2013.09—2015.07. 哈尔滨工业大学 软件工程

** 工作实习经历：**

2013.12—2014.02中软国际ETC实训基地“网络监控器、Linux内核”项目实训关于目录相关操作命令解析部分功能的C\C++代码实现，应用STL库对基本目录部分指令部分进行编写，以实现子进程下的对目标内容的查看，创建，删除，修改权限等基本操作。

2014.6—2015.5 中科院信息工程研究所第二研究室网络安全工程项目组工作实习，主要自主开发了C语言基于PKI和硬件USBkey的身份认证和加密通信模块。参与钓鱼网站检测（C++开发）的项目部分模块部分的开发和维护并做相关的测试和shell、python脚本的编写，参与APT攻击检测项目的相关测试。

**课程设计：**

2012.11—2013.01基于西门子S-700的PLC相关程序设计：交通灯系统的控制，物料传送系统，钻头深孔钻探周期循环控制程序，

2013.04—2013.07 毕业设计：“基于柔性薄膜传感器的触觉检测仪研制”，基于

51单片机C语言实现A/D转换后的压电信号处理部分，实现幅值异步变化显示。

2013.10—2013.11基于MVC模式下的‘枪支销售系统’JavaWeb程序设计，使用Html+CSS，JavaScript，基于Ajax的用户界面设计与基本信息处理部分。