doi:10.3969/j.issn.1672-1292.2021.01.005

网站变更监测预警系统的设计与实现

何诗佳[1,2],刘晓强[1],李柏岩[1],蔡立志[2],胡　芸[2]

(1.东华大学计算机科学与技术学院,上海 201620)
(2.上海市计算机软件评测重点实验室,上海 201112)

[摘要]　网站易成为黑客入侵篡改的对象,网站的实时变更监测对于网站安全尤为重要. 针对目前大规模进行网站实时变更监测的难点,设计并实现了一种基于非关系型数据库和消息机制的网站变更监测方案. 系统采用爬虫技术进行网站页面实时爬取,通过分布式数据存储和消息机制实现对多网站的实时分析,采用了 MD5 值与文本对比相结合的算法进行网站内容变更监测,并对监测结果进行可视化. 此外,当网站出现异常变更时,支持实时处理告警及紧急切断服务,减少由于网站内容被篡改所带来的不良影响.

[关键词]　网站内容篡改,网站变更监测,MD5,文本对比算法,分布式存储,消息机制
[中图分类号]TP391.1　[文献标志码]A　[文章编号]1672-1292(2021)01-0030-06

Design and Implementation of Website Change Monitoring and Early Warning System

He Shijia[1,2],Liu Xiaoqiang[1],Li Baiyan[1],Cai Lizhi[2],Hu Yun[2]

(1.College of Computer Science and Technology,Donghua University,Shanghai 201620,China)
(2.Shanghai Key Laboratory of Computer Software Testing and Evaluating,Shanghai 201112,China)

Abstract:Websites are easy to become the target of hacking and tampering. The real-time monitoring of website changes is particularly important for the safety of websites. Regarding the difficulties of large-scale real-time website change monitoring,we design and implement a website change monitoring system based on non-relational database and message mechanism. It uses crawler technology to crawl web pages in real time,and realizes real-time analysis of multiple websites through distributed data storage and message mechanism. An algorithm combining MD5 value and text comparison is designed to monitor website content changes and the results are visualized on the monitoring browser. When abnormal changes occur,it supports real-time alarming and emergency cut-off services in order to reduce the adverse effects caused by website content tampering.

Key words:website content tampering,website change monitoring,MD5,text comparison algorithm,distributed storage,message mechanism

网站是团体和机构必不可少的信息发布和交流平台,易成为黑客攻击的对象. 黑客攻击网站时篡改网站内容,篡改后的页面往往会出现一些不良信息,对机构形象和社会安定产生负面影响. 网站变更监测的目标是对网站页面内容进行对比监测,实时报告变更情况,便于监控人员及时发现页面中的篡改内容,减少网站篡改带来的损失,维护网站安全.

网站变更监测存在着许多问题和难点. 实验表明,网站变化的频率与域名种类、页面大小等因素密切相关,大约有 40%的网站在一周内发生了变化,大约 50 天内一半的网站都发生了变化,尤其是.com 网站,一天内就有 25%的网站发生了变化[1]. 网络中的网站数量庞大以及时刻动态变化的特性,使得对网站变化的监测十分困难. 同时,为了监测网站变化,反复重载所监测的页面进行比较,对系统的并行性要求很高. 基于目前网站变更监测的难点,本文设计和实现了一种基于非关系型数据库和消息机制的网站变更监测方案,可满足大规模实时监测需求. 在变更监测算法上采用 MD5 值比较与基于文本比较算法的结合,

收稿日期:2020-08-08.
通讯作者:刘晓强,教授,研究方向:智能信息系统、语义 Web、软件工程. E-mail:liuxq@dhu.edu.cn

检测精度高,能够定位到变更的具体位置. 此外,系统增加了监测预警功能,可实现监测实时处理告警及紧急切断服务支持.

1 网站变更监测相关研究工作

网站变更监测的方案主要分为两类:基于网站服务端的本地监测和基于客户端的远程监测.

基于网站服务端的本地监测主要采用事件触发技术、核心内嵌技术、外挂轮询技术等方法,其优点是可实时防护、技术精度高,但需在每个服务器上安装专门的软件,占用了服务器的系统资源,管理员操作复杂,不适合大规模的监测[2].

基于客户端的远程监测只需知道网站的域名,适合大规模多网站的实时监测服务. 远程监测网站变更主要采用以下算法:

(1)基于 HTTP 协议头的状态监测

基于 HTTP 协议头的状态监测常用属性有 Last-Modified 和 ETag. Last-Modified 中记录了网页在服务器端最后被修改的时间,ETag 中记录了服务器根据网页资源所生成的标记号[3]. 服务器通过验证 Last-Modified 字段和 ETag 值即可判断网页内容是否发生变化. 该方法快速简单,可节省大量不必要的网络资源,适用于静态网页变更监测. 对于动态网页,Last-Modifed 对应服务器发送 Response 的时间并非网页的最后更新时间,ETag 通常为空值,该方法无效.

(2)基于网页的 MD5 值监测

MD5 算法即 Message-Digest Algorithm 5(信息-摘要算法 5),监测时通过比较前后网页的 MD5 值来判断页面内容是否发生变更. 该方法实现简单,但过于严格,如页面中存在统计访问人数或记录时间的脚本,一旦发生变化,也会导致 MD5 值的改变,而这些改变通常无意义. 此外,该方法无法定位到发生变更的具体位置.

(3)基于文本比较算法的对比监测

网页本质上是纯文本文件,可将网页看成一个长字符串,通过文本比较算法来监测两个网页之间的差异. 文本比较算法主要有基于文本的编辑距离(levenshtein distance,LD)算法和基于文本的最长公共子序列(longest common subsequence,LCS)算法. 基于文本比较的对比算法实现简单、检测速度快,但直接使用文本比较算法来计算网页间的字符差异效率非常低.

(4)基于网页结构的对比监测

基于网页结构的对比监测是指根据网页代码生成一棵 DOM 树,采用遍历和树节点一一比对的方法来定位网页间的差异[4]. 其优点是能够全面比对网页的内容、结构、样式,适合只关注网页某个部分的监测,允许用户定制. 该方法不适合结构复杂的页面,会导致 DOM 树庞大,从而效率低、准确性低.

本文基于客户端的远程监测系统,将 MD5 值比较与基于文本比较算法结合以实现网站变更监测,该方法实现简单,检测精度高,能够定位到变更的具体位置. 由于传统关系型数据库不能满足对大量网站内容的快速查找,本系统采用非关系型分布式数据库 ElasticSearch 存储网页内容和变更结果[5],并采用高性能、易部署的 NSQ 消息队列实时处理不断新增的数据[6].

2 网站变更监测预警系统设计

本文设计的网站变更监测预警系统以 B/S 架构为核心,有 3 个核心模块,整体架构如图 1 所示.

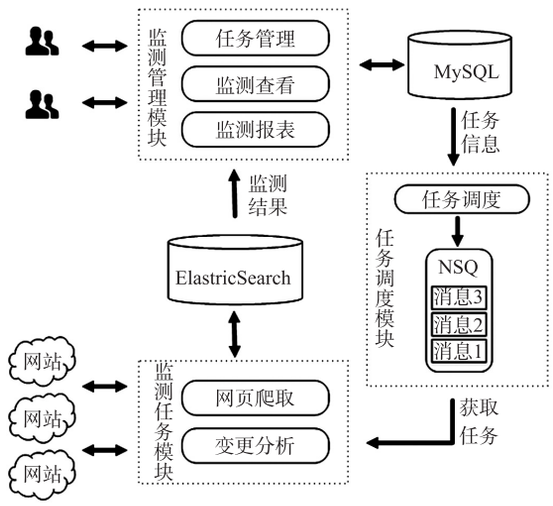

图 1 网站变更监测预警系统架构

Fig. 1 The architecture of website change monitoring and early warning system

2.1 监测管理模块

监测管理模块以 Web 网站形式向管理员提供

对目标网站的监测和管理服务,包括监测任务管理、监测结果查看和监测报表生成. 用户通过该模块可以添加监测网站、开启或停止监测任务,使用 MySQL 数据库存储用户信息及相关任务信息,从 ElasticSearch 数据库中取出最终的监测结果可视化展示在页面上并生成对应的报表便于管理员查看,管理员可针对异常情况发出关闭服务器命令.

2.2　任务调度模块

任务调度模块负责对监测任务模块进行配置并初始化 NSQ 消息队列. NSQ 消息队列可用于大规模系统中的实时消息服务. 当任务调度获取到网站监测任务开启或关闭的信息后,能够对任务的状态进行更新,通过 NSQ 消息队列将任务信息分发到监测任务模块,启动对应的爬虫任务或变更监测分析任务.

2.3　监测任务模块

监测任务模块分为网页爬取和变更分析两类. 监测任务开启后,爬虫定时去爬取被监测网站的网页信息并存储到 ElasticSearch 数据库中;变更监测分析任务通过对比前后两版页面信息将变更结果存入 ElasticSearch 数据库中,为最后以网页形式为用户呈现监测结果提供动态数据支持.

网站变更监测预警系统采用构件化技术,各模块独立运行,根据系统负载情况及用户需求适当开启监控任务,增强系统的稳定性与生命周期. 根据网站的变化判断是否正常,定义预警策略进行消息告警,并支持服务切断.

系统实现中,任务调度模块的任务调度器和监测任务模块的爬虫采用 Golang 语言开发;监测管理模块和监测任务模块的变更分析采用 Python 语言开发;网站 Web 端基于 Django 框架;采用非关系型数据库 ElasticSearch 存储网页内容和变更分析结果,其他信息存储在 MySQL 数据库中.

3　网站变更监测算法设计

3.1　数据获取及处理流程

用户通过监测任务管理模块开启监测任务后,经任务调度器发送任务信息给 NSQ 消息队列,爬虫任务启动后监听 NSQ 中的任务信息进行页面爬取,爬取到的页面内容存储在 ElasticSearch 数据库中. 对同一 URL 的网页,每次爬取页面时会生成对应的版本号,进行对比分析时可根据爬虫版本号每次从 ElasticSearch 数据库中取出当前 URL 所对应版本的内容与上一版本进行变更对比,对比结果存入 ElasticSearch 数据库中.

3.2　网站变更对比算法

网站变更分析采用 MD5 值进行初步比较,而后结合 LCS 算法进行文本内容比较.

3.2.1　采用 MD5 算法进行初步比较

一旦网页的 MD5 值发生变化,则网站的内容一定发生了变化. 本系统采用 Python 的 hashlib 库返回页面的 MD5 值进行比较. 若 MD5 一致,则表明页面内容未发生变化,不再进行下一步比较.

3.2.2　采用 LCS 算法进行文本内容比较

当对比页面 MD5 值不同时,采用 LCS 算法进行内容变更比较. LCS 算法中的子序列指不改变序列中元素的顺序,从序列中删除任意某些元素而获得的新序列,例如字符串 acdfg 与 akdfc 的最长公共子序列为 adf. 对于 LCS 问题的解决思路采用动态规划的方法.

《算法导论》第 3 版中通过构建矩阵实现该算法的求解[7]. 设所给的两个序列为 $X=<A,B,C,B,D,A,B>$ 和 $Y=<B,D,C,A,B,A>$. 由算法 LCS 计算出的结果 Z 为 $<B,C,B,A>$,求解过程如图 2 所示.

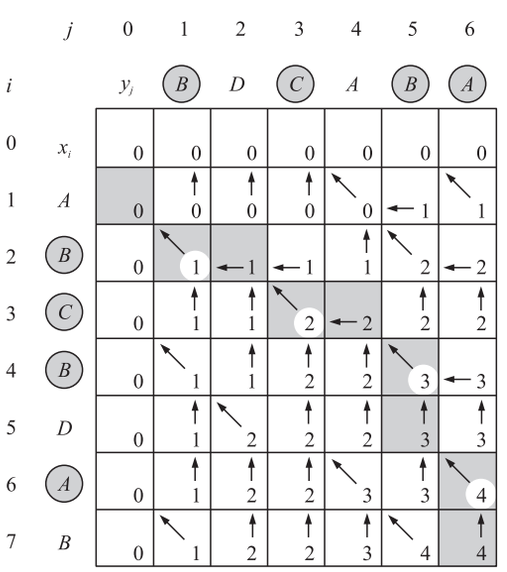

图 2　LCS 问题求解矩阵

Fig. 2　Sovling matrix of LCS

本文定义二维数组 $\boldsymbol{c}[i][j]$ 表示 X_i 和 Y_j 的 LCS 的长度,$\boldsymbol{b}[i][j]$ 中存放每次获得的解的方向. 算法的核心

思想如下:

```
1  Len1=序列 X 的长度
2  Len2=序列 Y 的长度
3  For i=0 to Len1
4      c[i][0]==0
5  For j=0 to Len2
6      c[0][j]==0
7  For i=1 to Len1
8    For j=1 to Len2
9    If   X[i]==Y[j]
10         c[i][j]==c[i-1][j-1] + 1
11         b[i][j]==1   // 1 表示箭头方向为 ↖
12   Else If c[i-1][j] >=c[i][j-1]
13           c[i][j]==c[i-1][j]
14           b[i][j]==2   // 2 表示箭头方向为 ↑
15   Else
16           c[i][j]==c[i][j-1]
17           b[i][j]==3   // 3 表示箭头方向为 ←
18 Return b,c
```

当得到完整的矩阵之后,通过倒推来得到相应的子序列. 从最后一个位置开始往前遍历 b 数组(i,j 代表当前字符在数组中的位置):

若 $b[i][j]==1$,则代表该字符是 LCS 的一员,存下该值后 $i-1$,$j-1$,继续向左上角查找;

若 $b[i][j]==2$,则代表该字符不是 LCS 的一员,$i-1$,向上查找;

若 $b[i][j]==3$,也代表该字符不是 LCS 的一员,$j-1$,向左查找.

在代码实现时,每次回溯根据矩阵箭头标示生成初步的对比坐标点,在合并对比结果时通过坐标位置判断内容为新增、删除或更改.

LCS 算法的时间复杂度和空间复杂度与进行比较的字符串长度成正比. 为了减少时间和空间耗费,本系统做出以下改进:

(1)在处理页面时,将页面以 html 标签"< >"切分为单位进行比较;

(2)使用碎块化对比,比对时若同一位置处两段切分完全相同,用一维数组记录下当前位标,并将位标增加 100 继续查找,否则位标加 1;按照数组中记录的下标对比较对象进行碎块化,分别进行 LCS 算法处理,与比较对象相同度越高时,对比速度越快.

3.3 变更结果处理及展示

通过网站变更算法处理,对爬虫后的网页源码进行标记,标记分为变更标记和类型标记. 以 html 标签为一个切分,监测到增加部分在切分代码前添加变更标记"+",减少部分添加变更标记"-",改变部分添加变更标记"?",未变部分添加变更标记"=". 通过 html 标签判断变更类型,若改变部分为图片,则添加类型标记"I";若改变部分为文本内容,则添加类型标记"T",其他类型标记为"N". 处理后的网页代码存储在 ElasticSearch 数据库中.

变更结果以网页形式实时展示,便于监控人员更直观地操作. 读取数据库中变更后的源码,根据标记加入相应的 span 标签,通过标签引入对应的 CSS 样式,最终展示在页面中.

图 3 所示为东华大学研究生系统任务中在 7 月内产生变更的网站及变更的具体信息. 记录显示了对应网站变更状态是否正常、变动范围、变更是否可见及变更各项的统计.

如图 4 所示,点击查看详情按钮,即可查看到页面中具体内容的变化位置,并可通过高亮颜色来增强可视化效果.

如图 5 所示,对 CSS、Javascript 的改变不直接显示在页面上,管理员可通过系统中的代码对比功能来

判断变化是否正常.

图 3　变更统计详情页面

Fig. 3　Page of change statistics

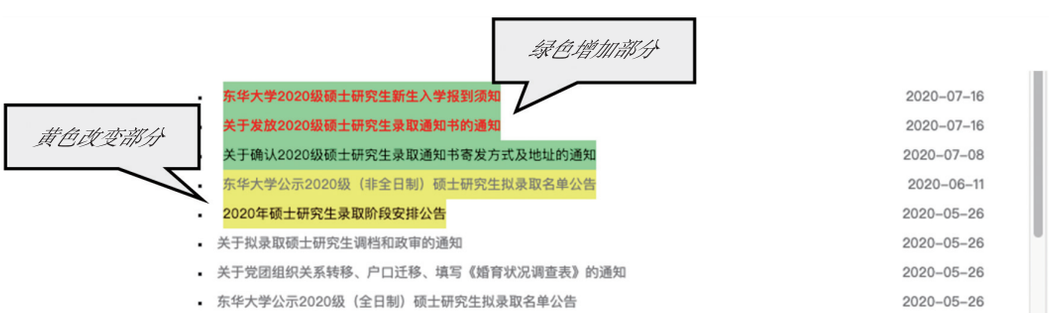

图 4　变更结果展示页面

Fig. 4　Page of change result

图 5　变更代码对比页面

Fig. 5　Page of code comparison

3.4　实时预警和切断服务

系统采用设置变更率阈值来判断变更是否正常. 变更率阈值为增加、删除、变化的内容占整个网页的比重. 如图 6 所示,系统默认设计了二级告警,当阈值小于等于 0.3 时,定义为普通消息,普通信息显示在系统主页中进行提示(图 6(a));当阈值大于 0.3 时,变更异常,定义为告警消息,系统会向客户单位发送

邮件进行告警(图 6(b)). 用户可通过定义变更率阈值来定义预警策略. 若管理员通过系统发现网站存在恶意变更内容,可紧急切断服务,待网站正常时恢复访问.

(a) (b)

图 6 预警消息

Fig. 6 Warning message

4 结论

网站易成为黑客攻击的对象,对网站变更的监测往往能够减少网站被篡改所带来的不良影响. 本文所设计系统实现了对网站变更情况的实时监测与预警,并支持对多网站的大规模监测需求,为网站内容安全监控提供了一种基础架构和解决方案.

目前,系统中还存在以下问题需要进一步研究和改进:

(1)对于 LCS 算法,由于回溯时左侧值等于上方值时默认向上回溯,最终只能得到一种结果,而实际上通过 LCS 算法回溯路径不同可获得多种结果. 在所有结果中,如何找出最符合网站变更情况的结果是下一步需要研究的方向;

(2)系统对 CSS 和 Javascript 的变更只能通过管理员代码对比来判断,而 CSS 和 Javascript 的改变往往会给网站带来巨大的影响,因此在 CSS 和 Javascript 的监测上还需要进行优化.

[参考文献](References)

[1] FETTERLY D,MANASSE M,NAJORK M,et al. A large-scale study of the evolution of web pages[J]. Software Practice and Experience,2004,34(2):213-237.

[2] 魏文晗,邓一贵. 基于局部变化性的网页篡改识别模型及方法[J]. 计算机应用,2013,33(2):430-433.

[3] 盛博文. WEB 网站内容更新检测关键技术研究[D]. 哈尔滨:哈尔滨工程大学,2017.

[4] 刘江. 网页篡改监控系统的设计与实现[D]. 北京:北京邮电大学,2018.

[5] 王伟,魏乐,刘文清,等. 基于 ElasticSearch 的分布式全文搜索系统[J]. 电子科技,2018,31(8):56-59,65.

[6] 陈付梅,韩德志,毕坤,等. 大数据环境下的分布式数据流处理关键技术探析[J]. 计算机应用,2017,37(3):620-627.

[7] THOMAS H C,CHARLES E L,RONALD L R,et al. 算法导论[M]. 3rd ed. 殷建平,徐云,王刚,等译. 北京:机械工业出版社,2013.

[责任编辑:严海琳]