计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(8):232−237 [doi: 10.15888/j.cnki.csa.005895]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

Web 渗透测试的信息抓取策略研究①

宋雅楠, 刘　萍

(陆军军官学院, 合肥 230031)

摘　要: 文章研究了在渗透测试中 Web 站点的信息抓取的问题. 针对 Web 渗透测试对于信息抓取的全面性、高效性的需求, 本文通过对 Web 站点进行研究分析, 提出了基于导航链接的网络爬虫策略, 并通过减少迭代次数对传统的 MD5 去重算法进行了改进, 优化了 URL 去重的效率. 通过实验验证表明, 该爬虫策略的信息抓取覆盖率和网页下载效率均有所提高.

关键词: 网络爬虫; Web 信息抓取; URL 去重; MD5 算法; 覆盖率

引用格式: 宋雅楠,刘萍.Web 渗透测试的信息抓取策略研究.计算机系统应用,2017,26(8):232–237. http://www.c-s-a.org.cn/1003-3254/5895.html

Research on Information Fetching Strategy of Web Penetration Test

SONG Ya-Nan, LIU Ping

(Army Officer Academy P.L.A, Hefei 230031, China)

Abstract: This paper investigates the problem of information crawling in Web site in penetration testing. In order to meet the requirement of high efficiency and comprehensiveness of information fetching in Web penetration test, in this paper, we researched and analyzed the Web site, and proposed a web crawler strategy based on navigation link. Besides, to optimize the efficiency of URL de-emphasis, we improved the traditional MD5 de-emphasis algorithm by reducing the number of iterations. The experimental results show that the coverage of information fetching and web page download efficiency are raised with the Strategy of Web crawler.

Key words: Web crawler; Web information extraction; URL de-emphasis; MD5 algorithm; coverage

1　引言

抓取 Web 站点中的海量数据是 Web 应用渗透测试的重要组成部分, 也是其实现的必要过程. 通常情况下, 这项工作大多采用网络爬虫技术来完成. 利用爬虫技术, 可以实现对检测目标信息搜集, 探索目标的结构特征和接口特征等功能, 发现与检测目标相关的资源信息, 确定渗透测试目标的范围, 从而为后续的渗透测试工作提供支持. 与搜索引擎不同, 应用于渗透测试的网络爬虫不过多关注抓取页面主题的相关程度, 我们更多的是把目光着眼于解决如何既保证高效又尽量全面的抓取 Web 信息的问题. 因此, 当前渗透测试普遍采取传统的索引爬虫来获取待检测 Web 站点的信息.

索引爬虫是通过提取一个给定起始页面中所包含的链接, 将其 URL 作为索引, 依据索引进行爬行, 在爬行过程中, 不断将新的 URL 加入索引队列, 循环往复, 从而抓取全部 Web 页面. 由于这种爬虫策略不考虑页面优先级, 只要是新的 URL 链接就进行爬取, 能够最大程度的实现对 Web 应用中信息的抓取, 虽然理论上保证了抓取的覆盖率, 但是随着 Web 站点中信息的海量增加, Web 应用规模的逐渐扩大, 索引爬虫的抓取效率早已不能满足需求, 已经成为了制约渗透测试效率的短板.

通过上述分析, 我们可以得出这样的结论: 当前应用于渗透测试系统的网络爬虫的技术瓶颈主要体现在

① 基金项目: 合肥市基本建设项目(2013CGFZ1948)
收稿时间: 2016-12-06; 采用时间: 2016-12-26

两个方面: 一是信息抓取的覆盖率, 渗透测试系统通过信息抓取的过程实现对待检测网络目标信息的收集, 因此信息抓取的覆盖程度将直接影响检测的全面性; 二是信息抓取的效率, 这也直接关系到渗透测试的效率.

为了解决渗透测试中网络爬虫在性能上存在的瓶颈, 国内外很多学者进行了相关探索. 彭赓和范明钰[1]讨论了一种应用于 SQL 注入漏洞的改进的索引爬虫技术, 他们对 URL 的筛选策略和流程进行了优化; Y. Huang[2]针对大部分 Web 应用安全漏洞存在于表单项当中这一特点, 提出了一种新颖的表单爬虫, 并将其应用于漏洞检测, 但是这种爬虫的抓取对象只局限于 Web 站点中的表单信息; 赵亭等人[3]对表单爬虫进行了改进, 他们利用页面中的导航信息来对 Web 应用系统中的表单接口进行抓取, 在一定程度上提高了漏洞检测的效率, 但是 Web 应用中的信息复杂且海量, 这种方法其抓取对象本质上还仅仅只是 Web 中的表单信息, 不能保证信息抓取的覆盖率. 对于渗透测试来说, 使用单一表单爬虫的信息抓取策略忽略了 Web 站点中其他因素带来风险的可能性, 不能保证渗透测试的全面性.

为了解决渗透测试过程中信息抓取的相关问题, 平衡网络爬虫高覆盖率和低抓取效率之间的矛盾, 本文在现有相关研究的工作基础上, 针对已有方案的不足之处, 提出了一种改进的 Web 信息抓取策略. 首先通过大量分析观察, 总结了 Web 站点的页面特点, 并基于该特点改进了基于导航链接的扫描策略, 然后在此基础上, 通过减少运算中的迭代次数, 详细阐述了对传统 MD5 算法的优化思路, 对抓取过程中的 URL 去重方案进行了改进. 最后, 通过实验对 Web 抓取策略的抓取覆盖率和改进算法的效率进行了验证.

2 改进的网络爬虫策略

2.1 基于导航链接的页面扫描策略

Web 应用系统大多采用框架式的模块设计, 为保证页面内容充实并且丰富美观, 开发人员在进行网页设计时, 往往加入大量图片链接或是多种编程语言设计生成的展示文档等来填充页面, 这就导致了我们看到的网页并不像普通文档那样简洁, 这样在网络爬虫进行抓取时会存在大量的干扰信息, 给信息抓取工作带来了很大的挑战[4].

图 1 为某著名门户网站的首页, 我们通过对网站的页面布局进行观察分析可以得出: 方框 1 内的链接一般来说导向门户网站的二级、三级页面, 具有导航性质, 相对于方框 2 内的链接而言, 它所包含的信息更为丰富, 也就是说, 其潜在的不安全因素也更多. 通过分析对比方框 1 和方框 2, 我们发现导航链接在显示上与普通链接是有明显区别的——为了吸引用户点击, 导航链接在页面上的显示更为突出. 链接中的锚文本字数一般比较简短、内容十分具有概括性且字体被加粗或是字体颜色被重点标出等; 导航链接的链接组在分布格局上来看更加整齐、排列具有连续性、并且在页面中的位置更为醒目等.

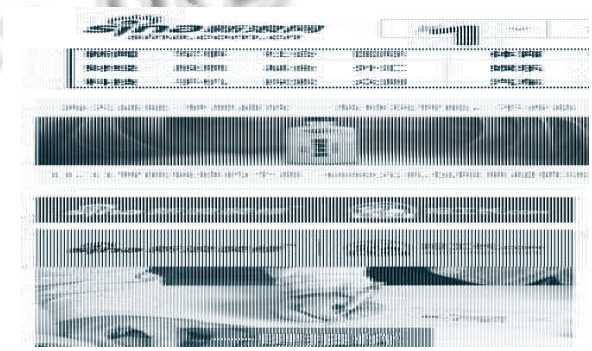

图 1 某网站导航链接示意

通过对大量类似 Web 站点进行观察分析, 我们可以总结出页面中具有导航性质的链接往往包含着大量的信息. 这些导航链接不仅指向性明显, 在分布上也是有规律可循的:

(1) 在页面中反复出现.

(2) 在页面中的显示格式和其它链接有明显不同, 一般突出显示或呈现规则性, 主要表现在: 链接文字的字体与默认字体不同、文字颜色或样式突出显示, 锚文本长度简短, 链接的位置在页面中居上或居左等.

因此, 我们在寻找判定导航链接时就以此为标准, 分为三步, 如图 2.

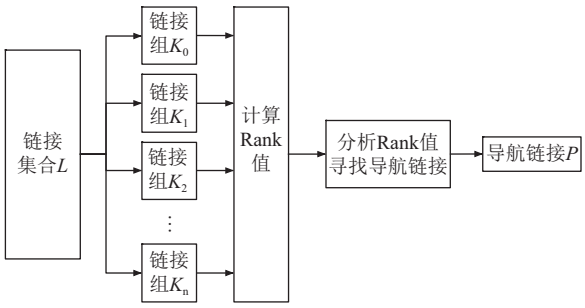

图 2 导航链接的判定

步骤一. 对页面中的链接进行分组和排序. 浏览器可以获取到网页的 html 源码, 经过浏览器内核 HTML 解析引擎对 html 源码进行解析, 能够得到排版引擎的相关数据, 从而获得某个网页元素的具体坐标. 利用浏览器的这项功能, 我们首先找出页面中的所有链接, 记为集合 L. 然后通过浏览器排版引擎获得链接在页面内的位置信息, 将超过 3 个的连续水平或垂直分布的链接划分为一组, 记为{k0, k1…Kn}, 这样集合 L 被划分为无交叉的多个链接组.

步骤二. 利用体现页面分布规律的式(1)计算每个链接组 K 的 Rank 值.

$$Rank(k)=font_{color(k)}\times W(c)+font_{family(k)}\times W(f)+size(k)+\frac{anchorttext(k)}{total_{text}}+\frac{1}{dist\_top(k)}+\frac{1}{dist\_left(k)} \quad (1)$$

在上式中, $font_{color(k)}$表示链接组 K 中锚文本文字的颜色, $W(c)$为其判断值. 若锚文本文字为黑色, 则表示该链接没有被突出显示, $W(c)$值为 0, 否则为 1; $font_{family(k)}$ 表示链接组 K 中的字体样式, $W(f)$为其判断值. 若字体为默认宋体, 则值为 0, 否则为 1; $size(k)$是链接组 k 中包含链接的个数; $anchorttext(k)$和 $total_{text}$ 分别是链接组中锚文本字数与页面总字符数, 其比值越接近 1, 则表明该链接组中锚文本字数越简短, 它的重要程度也就越高; $dist_top(k)$和 $dist_left(k)$分别代表着链接组距页面上沿和页面左沿的距离, 页边距越小越具有导航性质, 为方便统一计算, 这里我们取倒数.

步骤三. 计算完成后, 将 Rank 值由高到低进行排序, 取出排名前三位链接组中的链接, 记作 P, 顺着链接 P 访问其导向的页面, 将该页面记作 Q, 若在 Q 页面中也存在链接 P, 则可判定 P 为导航链接. 按照这种方法进行链接重现分析, 直至将所有链接组判断完毕.

2.2 基于改进 MD5 算法的 URL 去重策略

通过上一小节的分析, 我们可以知道反复出现是导航链接的主要特点之一, 所以不可避免的会带来重复抓取的现象, 这种现象不仅浪费 CPU 资源致使爬虫的效率降低, 还极有可能使系统陷入死循环中. 因此在爬取前对 URL 进行去重是一个不可或缺的环节. 一个优秀的 URL 去重模块应该具备较高的查找速度, 较低的碰撞率和误判率, 同时为了节省系统资源, 应该尽量减小其时间复杂度和空间复杂度. 基于以上考虑, 本文利用 MD5 加密算法的指纹碰撞率低的特点, 确定出了基于 MD5 改进算法的 URL 去重策略.

传统的 MD5 算法思想是将输入的信息进行 0、1 填充至 512 的整数倍, 在填充过程中用后 64 位表示信息的原始长度. 然后以每 512 位为一组进行分组处理. 接着将每一分组再划分为 16 个 32 位子分组. 通过进行 64 次算法定义的级联循环位操作(其中主循环 4 轮, 每轮 16 次), 得出一个 128 位的散列值, 该散列值即为输入文本的 MD5 指纹信息.

根据生日悖论, 理论上两个不同的 URL 链接经过 MD5 运算后, 其指纹信息发生碰撞的概率仅为$\frac{1}{2^{64}}$, 这基本能够保证不同 URL 链接产生的指纹信息不重复, 可以较好的保证去重效果[5]. 但是 Web 站点根据其规模大小的不同, 可能包含的种子链接数量能够到达百万级甚至更大, 这样一来网络爬虫需要抓取的 URL 地址量就会剧增, 如果逐一对这些 URL 链接进行传统的 MD5 运算, 那么运算复杂度将会呈指数增加, 过多的系统资源将被浪费在 MD5 运算上, 渗透测试效率将会大大下降, 因此我们考虑通过减少 MD5 运算的迭代次数来提高去重模块的效率.

2.2.1 MD5 算法简述

首先, 对待处理的 URL 进行信息长度填充并进行分组 Y[0, 1, 2…N-1], 且将每一分组划分为 16 个 32 位子分组 M[0, 1, 2…15].

然后, 设定四个 32 位整数参数 : A=0x67452301, B=0xefcdab89, C=0x98badcfe, D=0x10325476.

以及四个按位操作的非线性函数, 其中 a, b, c 均为 32 位整数:

$$F(a,b,c)=(a\&b)|((\sim a)\&c)$$
$$G(a,b,c)=(a\&c)|(b\&(\sim c))$$
$$H(a,b,c)=a\hat{}b\hat{}c$$
$$I(a,b,c)=b\hat{}(a|(\sim c))$$

除此之外, 用 Mj 表示消息的第 j 个子分组(从 0 到 15), $<<<s$ 表示循环左移 s 位, t[i](i 取值从 1 到 64)为 MD5 算法给定常数, 则四种循环位操作分别为:

$$FF=a=b+((a+(F(b,c,d)+M[j]+t[i]))<<<s)$$
$$GG=a=b+((a+(G(b,c,d)+M[j]+t[i]))<<<s)$$
$$HH=a=b+((a+(H(b,c,d)+M[j]+t[i]))<<<s)$$

接着, 按照上述操作对输入运算进行 64 轮 MD5 循环位操作.

2.2.2 MD5 算法的优化思路

如果按照 2.2.1 小节中传统的 MD5 算法, 每个分组 $Y[N]$将进行 64 次循环位运算, 那么计算一个 URL 的 MD5 指纹将进行 64(N+1)次循环运算. 虽然应用传统的 MD5 算法去重效果更好, 但是假如一个 Web 站点包含上百万数量级的 URL 链接, 这么庞大的运算量将给 CPU 带来难以想象的负荷, 这对于渗透测试的整体测试效率而言是得不偿失的. 因此, 我们考虑通过减少运算的迭代次数来提高 MD5 算法的运算效率.

在每轮循环中, 分别以不同的顺序使用分组 $Y[i]$中的 4 个子分组 $M[j]$进行运算:

第一轮: 子分组 $M[j_1](j_1$=1, 2, 3, 4)参与第一个非线性函数$F(x,y,z)=(x\&y)|((\sim x)\&z)$的运算.

第二轮: 子分组 $M[j_2](j_2$=6, 7, 8, 9)参与第二个非线性函数$G(x,y,z)=(x\&z)|(y\&(\sim z))$的运算.

第三轮里: 子分组 $M[j_3](j_3$=11, 12, 13, 14)参与第三个非线性函数$H(x,y,z)=x\hat{}y\hat{}z$的运算.

第四轮里: 子分组 $M[j_4](j_4$=0, 5, 10, 15)参与第四个非线性函数$I(x,y,z)=y\hat{}(x|(\sim z))$的运算.

那么一个分组 $Y[N]$的循环过程将缩短至 16 步, 如表 1 所示.

从表 1 中可以看出, 在每一轮的运算过程中, 改进的 MD5 算法减少了对相同子分组的重复计算, 将整体的循环操作步数减少到了 16 步. 根据对传统 MD5 算法进行时间分析, 其复杂度是线性的 O(n). 而改进后 MD5 算法也是一种线性运算, 虽然其时间复杂度在量级上和传统 MD5 算法相比没有变化, 但其内部迭代算法已被减少 3/4 轮, 因此一定程度上提高了运算的效率. 同时改进后的 MD5 算法, 其每步操作的循环位移量也是唯一的, 通过这一点最大程度的将碰撞率维持在较低水平. 在第三小节的实验中我们可以看到改进的 MD5 算法效率已经有了明显的提升.

2.3 使用改进 MD5 算法的去重策略

运用 2.2.2 小节中改进的 MD5 算法生成 URL 链接的 MD5 指纹信息, 同时维护一个待抓取队列 unvisited 来存储待抓取的 URL 链接, 和一个 visited 链表队列存储已爬取页面的 URL 链接和该 URL 的 MD5 数值, 链表队列的数据结构如图 3 所示.

在进行 URL 去重时会使用大量的查找和插入操作, 上述链表队列这种数据结构能够满足算法中易查找易插入的需求, 能够合理地利用相应的内存来达到去重的目的.

表 1 MD5 改进算法的循环过程

/* 第 1 轮 * /	/* 第 2 轮 * /
FF (d, a, b, c, M[1], 12, 0xe8c7b756) //对应原算法第2步	GG (d, a, b, c, M[6], 9, 0xc040b340) //对应原算法第8步
FF (c, d, a, b, M[2], l7, 0x242070db) //对应原算法第3步	GG (c, d, a, b, M[7], 14, 0x676f02d9) //对应原算法第31步
FF (b, c, d, a, M[3], 22, 0xclbdceee) //对应原算法第4步	GG (b, c, d, a, M[8], 20, 0x455a14ed) //对应原算法第28步
FF (a, b, c, d, M[4], 7, 0xf57c0faf) //对应原算法第5步	GG (a, b, c, d, M[9], 5, 0x21e1cde6) //对应原算法第25步
/* 第 3 轮 * /	/* 第 4 轮 * /
HH (c, d, a, b, M[11], 16, 0x6d9d6122) //对应原算法第35步	Ⅱ(a, b, c, d, M[0], 6, 0xf4292244) //对应原算法第49步
HH (d, a, b, c, M[12], l1, 0xe6db99e5) //对应原算法第46步	Ⅱ(b, c, d, a, M[5], 21, 0xfc93a039) //对应原算法第52步
HH (a, b, c, d, M[13], 4, 0x289b7ec6) //对应原算法第41步	Ⅱ(c, d, a, b, M[10], 15, 0xffeff47d) //对应原算法第55步
HH (b, c, d, a, M[14], 23, 0xfde5380c) //对应原算法第3步	Ⅱ(d, a, b, c, M[15], 10, 0xfe2ce6e0) //对应原算法第58步

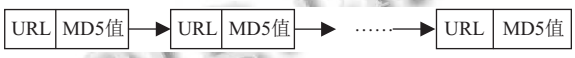

图 3 存储已爬取 URL 信息的链表队列 visited

综上所述, 一个基于改进 MD5 算法的 URL 去重策略的网络爬虫遵循下述算法:

a) 将选定的一系列种子 URL 加入 unvisited 队列.

b) 判断 unvisited 队列是否为空, 若非空, 跳转至下一步, 否则转向步骤 h).

c) 取出 unvisited 队列的队首 URL, 利用 MD5 生成器计算出其 MD5 指纹信息.

d) 将该 URL 的 MD5 指纹信息作为对比项, 遍历 visited 队列, 找出是否存在冲突. 若存在, 则舍弃该 URL, 转向步骤 b). 否则转向步骤 e).

e) 根据导航链接的判断公式, 找出该 URL 指向页面中是否包含导航链接, 若包含, 转向步骤 f); 否则转向步骤 g).

f) 将符合导航链接特征的 URL 放入 unvisited 队列.

g) 下载当前网页, 并将 URL 及其 MD5 指纹信息放入 visited 队列, 跳转至步骤 b).

h) 结束.

3 实验论证分析

为了验证本文提出的信息抓取策略和改进爬虫算法的性能, 搭建了如下的测试环境, 考虑到基于导航链接的信息抓取策略在不同主题特征的 Web 应用系统中抓取效果可能会不同, 所以为了验证导航链接策略的正确性, 在进行实验目标选择时, 我们选取了政企网络办公、电子商务、校园网论坛、信息资讯、网络资源存储等多个不同类型的 Web 应用系统作为信息抓取实验对象, 具体测试环境参数如表 2 所示.

表 2 测试环境

硬件环境	软件环境
处理器: Intel Core i7, 主频2.5 GHz, 内存: 6 GB, 硬盘: 1 T SSD网络: 1000 M以太网卡	Windows 7

本文研究的 Web 信息抓取方案主要对两个方面进行了改进: 基于导航链接的扫描策略和基于改进 MD5 算法的 URL 去重策略, 评价这两个策略好坏的指标分别是抓取覆盖率和抓取效率. 因此, 本小节我们通过对比改进爬虫、表单爬虫和索引爬虫的抓取结果, 先验证了导航链接策略的正确性和改进 MD5 算法的效率, 进而对抓取覆盖率和抓取效率进行了实验验证.

(1) 抓取覆盖率: 由于改进的 Web 信息抓取策略是利用导航链接进行 URL 扫描的, 因此导航链接的选取方案是否具有普适性对网络爬虫抓取信息的覆盖率有直接影响, 进而可能影响 Web 应用安全测试的检测结果是否全面. 在实验中我们分别根据 Web 应用系统主题的相关性不同, 分别对政企网络办公、电子商务、校园网论坛、信息资讯、网络资源存储五种不同的 Web 站点进行网络爬虫测试, 将爬虫获取到的页面数量与目标站点中所有页面数量的比例定义为抓取覆盖率, 同时使用表单爬虫和传统的索引爬虫作为对比项.

(2) 抓取效率: 在实验中我们通过将大约 43 亿条 URL 链接作为输入, 分别进行传统的 MD5 运算和改进的 MD5 运算, 测试运算效率. 此外, 使用本文提出的基于改进 MD5 算法的导航链接爬虫对上述五种不同的 Web 站点进行抓取实验, 记录抓取所用的时间, 同时使用表单爬虫和传统的索引爬虫作为对比.

(3) 性能对比: 针对应用系统, 分别使用传统的索引爬虫、表单爬虫和本章的改进爬虫进行定时爬取, 根据抓取结果对三种爬虫的性能进行对比分析.

1) 使用导航链接策略的抓取覆盖率分析

通过抓取覆盖率实验, 我们看出传统的索引爬虫根据其抓取策略可以抓取到每个测试站点几乎全部的页面; 而表单爬虫由于只抓取包含表单项的表单接口, 抓取覆盖率明显低于其他两种爬虫; 由于网络资源存储类型的 Web 站点页面分布规律性不强, 在使用导航链接策略的改进爬虫对该类型 Web 进行抓取时覆盖率欠佳, 在对其他四种类型的 Web 站点进行抓取时, 改进爬虫的覆盖率可以保持在百分之九十以上. 该实验证明: ①导航链接策略基本具备普适性. ②使用该策略的网络爬虫的抓取覆盖率基本可以达到预期目标. 具体实验结果如图 4 所示.

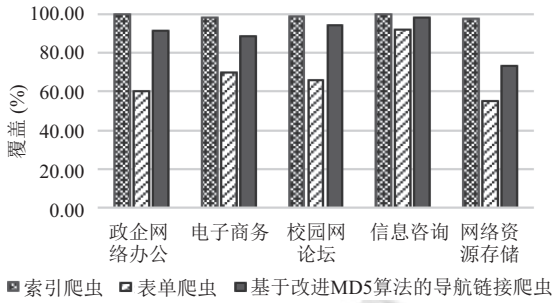

图 4 不同爬虫抓取各类型网站信息的抓取覆盖率

2) 抓取效率分析

根据统计数据显示, 每个 URL 链接平均所占字节数大约是 64 B, 为方便计算, 我们选取了 4 GB, 大小约等于 43 亿个 URL 链接的数据量作为实验数据, 分别进行传统和改进的 MD5 运算, 经过 1000 次运算后, 取平均耗时进行对比. 通过实验可以看出改进的 MD5 算法在运算时间上有明显优势. 如表 3 所示.

表 3 改进 MD5 算法和传统 MD5 算法对比表

计算方式	耗时(s)
传统MD5计算方式	22.5
改进MD5计算方式	16.3

此外, 使用本文提出的基于改进 MD5 算法的导航链接爬虫对五种不同的 Web 站点进行抓取实验, 记录抓取所用的时间, 同时使用表单爬虫和传统的索引爬虫作为对比, 由于各类 Web 站点的规模不同, 相互之

间不形成对比. 实验显示, 在单独对每一种类型的网站进行抓取时, 改进爬虫在抓取效率上比索引爬虫均有明显优化. 而表单爬虫由于其只对表单进行抓取的特点, 抓取效率对 Web 站点中表单数量的多少依赖程度很大, 在表单内容丰富的 Web 应用系统中, 采取改进 MD5 算法的导航链接爬虫的抓取效率要优于表单爬虫. 综合来看, 本文提出的改进爬虫在抓取效率上来看存在一定优势. 具体实验数据如图 5 所示.

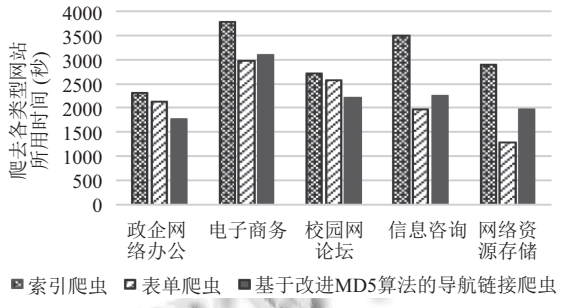

图 5 不同爬虫抓取各类型网站信息的抓取效率

3) 爬虫性能对比

综合上述实验, 为了将三种爬虫的性能做一个直观的对比, 我们针对合肥市基本建设项目收费网上办理应用系统分别使用三种爬虫进行了定时的抓取, 结果如图 6 所示.

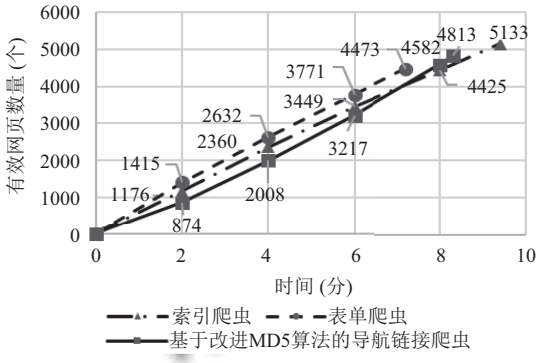

图 6 三种爬虫的性能对比

图 6 中结果表明, 在爬取的初期表单爬虫获取页面速度最快, 但由于其只抓取表单的策略, 爬行覆盖率不高; 采用广度优先策略的索引爬虫覆盖率虽然最高, 但是抓取的有效网页数量和爬行速度均被本章所设计爬虫超越, 其原因在于抓取的重复 URL 增多, 没有一个好的去重策略将严重消耗系统资源. 反观基于 MD5 改进算法的导航链接爬虫, 既能取得较高的覆盖率, 爬行效率也比较乐观.

4 结束语

本文分析了利用渗透测试进行 Web 安全检测时对信息抓取的需求特点, 在总结前人的工作基础上, 对网络爬虫的爬行策略和 URL 去重两个方面进行了优化. 通过对大量 Web 站点进行研究分析, 提出了基于导航链接的爬虫策略并对传统的 MD5 去重算法进行了改进, 使得网络爬虫抓取的页面覆盖率和抓取效率有了一定程度的提高, 能够较好的保证 Web 安全检测的全面性和检测效率. 最后, 通过实验对 Web 抓取策略的抓取覆盖率和改进算法的效率进行了验证, 实验证明了导航链接策略正确且基本具备普适性, 改进的 MD5 算法在耗时上有了明显缩短, 同时, 在该策略下爬虫既能取得较高的覆盖率, 爬行效率也比较乐观, 爬虫性能相对于传统爬虫有较大改进, 达到了提高信息抓取性能的目的.

参考文献

1 彭赓, 范明钰. 基于改进网络爬虫技术的 SQL 注入漏洞检测. 计算机应用研究, 2010, 27(7): 2605–2607.

2 Huang YW, Tsai CH, Lin TP, *et al*. A testing framework for Web application security assessment. Computer Networks, 2005, 48(5): 739–761. [doi: 10.1016/j.comnet.2005.01.003]

3 赵亭, 陆余良, 刘金红, 等. 基于表单爬虫的 Web 漏洞探测. 计算机工程, 2008, 34(9): 186–188, 215.

4 王欣. WEB 应用系统安全检测关键技术研究[博士学位论文]. 北京: 北京邮电大学, 2011.

5 严磊, 丁宾, 姚志敏, 等. 基于 MD5 去重树的网络爬虫的设计与优化. 计算机应用与软件, 2015, 32(2): 325–329, 333.