计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(1):75−80 [doi: 10.15888/j.cnki.csa.006770]

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

应用于拟态 Web 服务器的相似度求解方法①

王　灿, 倪　明, 喻卫东, 黎　想

(华东计算技术研究所, 上海 201808)
通讯作者: 王　灿, E-mail: canwcumt@foxmail.com

摘　要: 拟态 Web 服务器中表决器通过计算并比较异构执行体响应网页的相似性来判断响应是否为合法输出, 达到网页防篡改的目的. 目前表决器中将网页整体作为字符串输入, 采用字符串编辑距离方法计算网页的相似性, 存在计算量大忽略网页原有结构信息等问题. 本文采用改进简单树匹配方法, 通过对网页 DOM 树匹配判别得到网页的相似性, DOM 树节点匹配程度由节点字符串的编辑距离决定. 将本文算法应用于拟态 Web 服务器上, 进行网页篡改实验验证, 与现使用算法相比, 本文所采用算法在适应执行体异构性的基础上, 提高了表决器的计算效率和准确性.

关键词: 拟态 Web 服务器; 编辑距离; 简单树匹配; 相似性; 网页防篡改; DOM 树

引用格式: 王灿,倪明,喻卫东,黎想.应用于拟态 Web 服务器的相似度求解方法.计算机系统应用,2019,28(1):75–80. http://www.c-s-a.org.cn/1003-3254/6770.html

Similarity Calculation Method Applied to Mimic Web Server

WANG Can, NI Ming, YU Wei-Dong, LI Xiang

(East China Institute of Computing Technology, Shanghai 201808, China)

Abstract: The voter in the mimic Web server calculates the similarity of the heterogeneous executor response webpage in order to judge whether the response is legal output and thus to prevent webpages tempering. At present, the voter treats entire Webpage as a string and uses the string edit distance to calculate the similarity of the webpages. In this way, it caused problems such as large amount of calculation, ignorance of the original structure information of the webpages, and so on. In this study, the improved simple tree matching method is used to calculate the similarity of the webpages by calculating the similarity of the DOM tree of the webpages. The matching degree of DOM tree node is determined by the editing distance of the node string. The proposed algorithm is applied to the mimic Web server to verify the webpage tamper. Compared with the existing algorithms, the algorithm used in this study not only adapts itself to the heterogeneous but also improves the efficiency and accuracy of the voter.

Key words: mimicry Web server; editing distance; simple tree matching; similarity; webpage tamper-proof; DOM tree

国家互联网信息中心 2016 年 12 月报告显示, 中国网站总量达到 475.4 万个. 网站给人们生活提供多种便利, 同时也遭受越来越多的安全威胁, Web 服务器作为网站承载平台, 已经成为了网络攻击中的主要目标. 为应对越来越严峻的网络安全挑战, 我国独立自主提出了拟态防御技术[1], 拟态防御技术使用动态调度策略切换等价异构执行体, 构造出动态异构冗余的拟态环境, 利用所构建环境的不确定性和非持续性切断网络攻击链.

拟态 Web 服务器[2]是拟态防御技术在网站系统中

① 基金项目: 国家重点研发计划 (2016YFB0800100)
Foundation item: National Key Research and Development Program of China (2016YFB0800100)
收稿时间: 2018-07-07; 修改时间: 2018-09-05; 采用时间: 2018-09-18; csa 在线出版时间: 2018-12-26

一个应用, 在拟态网站系统中, 对每个异构执行体输出结果合法性的判决是安全的前提, 表决器中的相似度求解算法则是表决器的核心内容.

目前现有拟态 Web 服务器的表决器中, 通过字符串编辑距离衡量异构执形体响应网页的相似度[3]. 但是拟态 Web 服务器的异构执行体在发挥防御能力的同时也一定程度上造成了不同平台响应的差异性, 其中很多差异并不会影响网页的输出效果, 却很大程度上干扰了响应网页之间相似性的判决结果. 本文采用改进简单树匹配算法计算异构执行体响应网页的相似度, 并应用于拟态 Web 服务器的表决器中, 提高了表决器的效率和准确性.

1 基于字符串编辑距离的相似度求解

字符串编辑距离是一种常用的字符串相似度指标. 通过一些操作编辑一个字符串, 使其变成另外一个字符串, 编辑的最少次数即为衡量两个字符串的相似度[4]指标. 在网页相似性比较、网页相关性排序以及快速模糊匹配等方面有很多应用.

1.1 字符串编辑距离求解

编辑距离是指原字符串 A 经过插入、删除、替换三种编辑操作, 变成字符串 B 所需要的最少编辑次数.

设有 2 个字符串 A 和 B: $A=a_1a_2\cdots a_m$, $B=b_1b_2\cdots b_m$. 式 (1) 构造了 A 与 B 的 $(m+1)\times(n+1)$ 阶匹配关系矩阵 LD, 矩阵的第 1 列是字符串 A, 第 1 行是字符串 B:

$$\mathrm{LD}_{(m+1)\times(n+1)}=\{d_{ij}\}(0\leqslant i\leqslant m,0\leqslant j\leqslant n) \tag{1}$$

匹配关系矩阵中的元素被称为单元, 按如下方式计算:

$$d_{ij}=\begin{cases} i, & j=0 \\ j, & i=0 \\ \min(d_{i-1,j-1},d_{i-1,j},d_{i,j-1})+a_{i,j}, & i,j=0 \end{cases} \tag{2}$$

匹配关系矩阵中元素 d_{mn} 即为字符串 A 和 B 之间的编辑距离, 用 ld 表示.

1.2 基于 LD 的相似度计算公式

字符串 A 和 B 的相似度可通过 ld 计算, ld 越小, A 和 B 越相似, 反之, 差异越大. 根据编辑距离求解 A 和 B 的相似度公式如下:

$$Similar(A,B)=1-\frac{ld}{\max(|A|,|B|)} \tag{3}$$

式中, ld 为字符串 A 和 B 之间的编辑距离, $|A|$和$|B|$分别表示 2 个字符串的长度. $Similar(A, B)$ 值越大, 说明字符串 A 和 B 越相似.

异构 Web 服务器对于网页的请求响应存在差异性, 通常情况下这种差异并不会影响输出, 但是利用字符串编辑距离求解网页相似度时, 却会干扰相似度的计算结果. 而且网页作为一种结构化的内容[5,6], 将网页转化成字符串利用编辑距离计算相似性时, 会跨越结构层级比较, 忽略网页原有结构信息, 可能计算结果相似, 但呈现的结果却有较大差异. 因此现有字符串编辑距离计算方法在拟态 Web 服务器系统应用中存在短板. 本文为适用拟态 Web 服务器的要求, 给出了对节点采用编辑距离比较相似性的改进简单树匹配计算方法.

2 基于改进简单树匹配的相似度求解

针对拟态 Web 服务器中采用字符串编辑距离处理网页字符串计算量大, 忽略原有网页结构信息等问题, 本文将异构执行体响应网页转化成保留原结构信息的 DOM 树, 利用改进简单树匹配算法[7]计算异构执行体响应网页的相似度. DOM 树的节点是响应网页部分内容, 为兼容异构执行体造成的差异性, 在比较 DOM 树的节点时, 计算节点间的编辑距离, 根据编辑距离与所设阈值的大小判定节点是否相似.

2.1 简单树匹配基本原理

令 S 和 T 为两棵树, i 和 j 分别为 S 和 T 上的节点. 定义 S 和 T 的匹配为映射 M, 节点对 $(i, j)\in M$, i、j 不是根节点. $S=(R_S, S_1,\cdots, S_m)$ 和 $T=(R_T, T_1,\cdots, T_n)$ 是两棵 DOM 树, R_S、R_T 分别表示子树 S 和子树 T 的根节点, S_i 和 T_j 为第 i 个和第 j 个第 1 层子树. 根据编辑距离判断 S、T 两棵树的对应节点是否匹配, 当 R_S 和 R_T 匹配时, S 和 T 最大匹配为 $M_{S,T}+1$, $M_{S,T}$ 是<$S_1, S_2,\cdots, S_m$>和<$T_1, T_2,\cdots, T_n$>最大匹配. $M_{S,T}$ 由动态规划算法求出, 步骤如下:

步骤 1. 若 S_m 和 T_n 最大匹配大于任意一个 S_m 和 $T_i(1\leq i<n)$ 最大匹配, 那么 $M_{S,T}$ 是<$S_1, S_2,\cdots, S_{m-1}$>和<$T_1, T_2, \cdots, T_{n-1}$>之间的最大匹配加上 S_m 和 T_n 的最大匹配.

步骤 2. 否则, $M_{S,T}$ 等于<$S_1, S_2, \cdots, S_{m-1}$>和<$T_1, T_2, \cdots, T_n$>之间的最大匹配或<$S_1, S_2, \cdots, S_m$>和<$T_1, T_2,\cdots, T_{n-1}$>之间的最大匹配相似.

2.2 节点相似度计算

拟态 Web 服务器异构执行体输出结果会存在一

定差异, 计算待匹配节点的编辑距离, 根据编辑距离差异程度判断是否在可接受范围内. 网页 DOM 树节点内容的字串量不大, 采用改进字符串编辑距离方法计算对应节点的相似度, 方法如下:

$$Similar(n1,n2)=1-\frac{ed}{ed+lcs+\min(|pref|,|stuff|)} \quad (4)$$

式中, $n1$、$n2$ 是 2 个节点内容字符串, $|pref|$、$|stuff|$是字符串 $n1$、$n2$ 最长公共前缀长度和最长公共后缀长度; lcs 是 $n1$、$n2$ 去掉$|pref|$、$|stuff|$后剩余部分 $n1'$、$n2'$的最长公共子串的长度; ed 为 $n1$、$n2$ 去掉$|pref|$和$|stuff|$后的编辑距离.

如果 $Similar(n1, n2)>K1$, 则判为相似, 否则判为不同.

2.3 基本简单树匹配算法

对树 S 和 T 第一层进行递归匹配, 得到最大匹配, 结果保存在 W 矩阵中, 根据矩阵 W 中的值计算矩阵 M 中的值. 算法如算法 1.

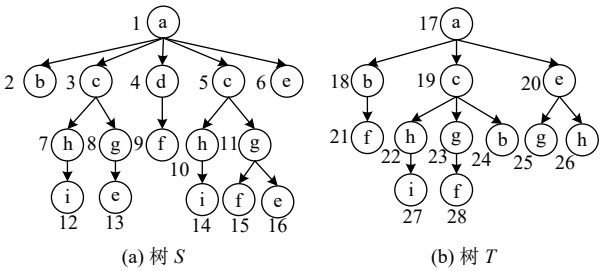
算法 1. 简单树匹配 STM(S, T)

输入: S, T
输出: 匹配的节点数

```
IF 树 S 和 T 的根节点不相似
RETRUN 0
ELSE
m=树 S 第一层节点数
n=树 T 第一层节点数
Initialize M[i, 0]=0 (i=0,…, m)
M[0, j]=0 (j=0,…, n)
FOR i=1:m
FOR j=1:n
M[i, j] = max(M[i, j–1], M[i–1, j], M[i–1, j–1]+W[i, j]);
W[i, j]=STM(Si, Tj);
ENDFOR
ENDFOR
RETURN M[m, n]+1
END
```

图 2 举例说明了基本简单树匹配算法执行过程. 为求图 1 中树 S 和 T 的最大匹配, 首先进行第一层子树的匹配, 定义 $M_{1\text{-}17}[5, 3]$是树 S 和 T 第一层子树的最大匹配; 由 $W_{1\text{-}17}$ 计算得到 $M_{1\text{-}17}$; 矩阵 $W_{1\text{-}17}$ 中的 $W[i, j]$表示 S 和 T 第一层第 i 个和第 j 个子树的最大匹配, 继续对 M 递归计算 W 值.

执行图 2 运算流程, 可以求出两棵树的匹配节点个数. 显然, 图 1 中 S、T 两棵树有 7 个节点匹配.



(a) 树 S (b) 树 T

图 1 两颗 DOM 树

	0	18	18-19	18-20
0	0	0	0	0
2	0	1	1	1
2-3	0	1	5	5
2-4	0	1	6	6
2-5	0	1	6	7

(a) $M_{1\text{-}17}$

	18	19	20
2	0	1	0
3	0	4	0
4	0	0	0
5	0	5	0

(b) $W_{1\text{-}17}$

	0	22	22-23	22-24
0	0	0	0	0
10	0	2	2	2
10-11	0	2	4	4

(c) $M_{5\text{-}19}$

	22	23	24
10	2	0	0
11	0	2	0

(d) $W_{5\text{-}19}$

	0	27
0	0	0
14	0	1

(e) $M_{10\text{-}22}$

	0	28
0	0	0
15	0	1
15-16	0	1

(f) $M_{11\text{-}23}$

	28
15	1
16	0

(g) $W_{11\text{-}23}$

图 2 部分节点匹配矩阵

2.4 相似度计算

拟态 Web 服务器网页防篡改应用中, 表决器根据异构执行体响应网页的相似度进行判决. DOM 树 T_1 和 T_2 相似度定义[8]如下:

$$similarity(T_1,T_2)=\frac{STM(T_1,T_2)}{(|T_1|+|T_2|)/2} \quad (5)$$

式中, $|T_1|$、$|T_2|$分别是两个树的节点数, STM(T_1, T_2) 是树 T_1 和 T_2 的最大匹配值. $similarity(T_1, T_2)$ 值越大, 表示网页 T_1 和 T_2 越相似.

3 拟态 Web 服务器防网页篡改应用

网站作为复杂信息系统, 漏洞无法避免. 常见 Web

服务系统包括 Web 服务器硬件漏洞、数据库漏洞、操作系统漏洞、网站源码漏洞等, 攻击者通常利用其中的一个或多个漏洞进行攻击.

3.1 拟态 Web 服务系统基本模型

拟态防御技术中动态异构冗余机制[9,10] (Dynamic Heterogeneous Redundancy, DHR) 使得攻击者无法建立稳定的攻击链接. 执行体的冗余使得即便某个执行体被攻击破坏, 也不会对系统的输出结果产生直接影响, 并且动态性保证了攻击结果无法重现, 大大降低了攻击者攻击成功的可能性.

拟态 Web 服务器借助动态异构冗余机制, 把 Web 服务系统部署在异构执行体上, 对输出结果的一致性进行择多判决后再输出, 实现抗攻击的目的.

图 3 是拟态 Web 服务系统架构图. 拟态 Web 服务系统由前端接入模块、Web 服务器池和控制器三部分组成. 前端输入模块主要实现了输入代理和输出代理两个功能, 是用户访问的实际入口和实际出口. 输入代理根据特定分发机制将用户请求分发至 Web 服务器池中的执行体上, 输出代理也被称为表决器, 根据特定判决算法对来自不同执行体响应进行表决输出. 池中包含多样、异构、冗余的执行体, 对外界提供 Web 服务. 实际使用中, Web 服务器池中只有一个执行体在线, 接收前端接入模块分发请求并做出回应; 池中其余执行体一直处于待机状态, 等待控制器模块的上线指令. 控制器模块根据系统异构性最大化策略调度池中的执行体, 降低了执行体的持续暴露时间和系统中存在一致性漏洞的可能性.

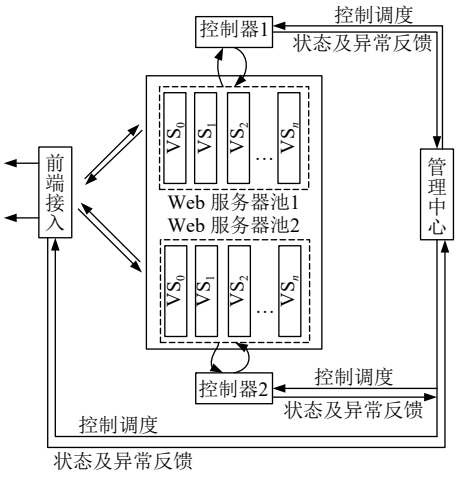


图 3 拟态 Web 服务器架构图

图 3 中的管理中心模块主要起到监测作用, 检测系统中其他模块运行状态, 处理各个模块的正异常信息, 保证拟态 Web 服务器正常运行.

3.2 求解方法在拟态 Web 服务器的应用

在拟态 Web 服务器的表决器中, 将响应网页解析成 DOM 树形式, 使 DOM 树除了包含网页展示的文本信息外[11,12], 还包含动态脚本信息. 对处理后的网页 DOM 树用改进简单树匹配方法求最大匹配, 计算相似度值.

计算两个网页的相似度值时, 采用递归方式对 DOM 树进行匹配, 求解树的待匹配节点的字符串计算编辑距离, 根据对应节点编辑距离的差异程度判断是否匹配. 统计出 DOM 树中节点匹配的个数.

在计算异构执行体 Web 服务器响应网页 T_1 和 T_2 的相似度之前, 将 T_1 和 T_2 作为普通字符串计算两者长度的比值, 并将计算结果与设置的阈值 $K2$ 进行比较, 若比值大于 $K2$, 则说明两个网页的差异较大, 直接判定为不同, 不予输出; 若比值小于等于阈值 $K2$, 表决器利用特定相似度计算算法进行相似度判决, 计算出 T_1、T_2 的相似度值后与所设置的阈值 $K3$ 进行比较. 若两个网页相似度大于等于 $K3$, 则说明两个网页差异在允许的范围内, 可判定为合法响应, 予以输出; 若相似度小于 $K3$, 则说明两个网页之间差异不在允许范围内, 判定为非法响应, 不予输出. 表决器执行流程如图 4 所示.

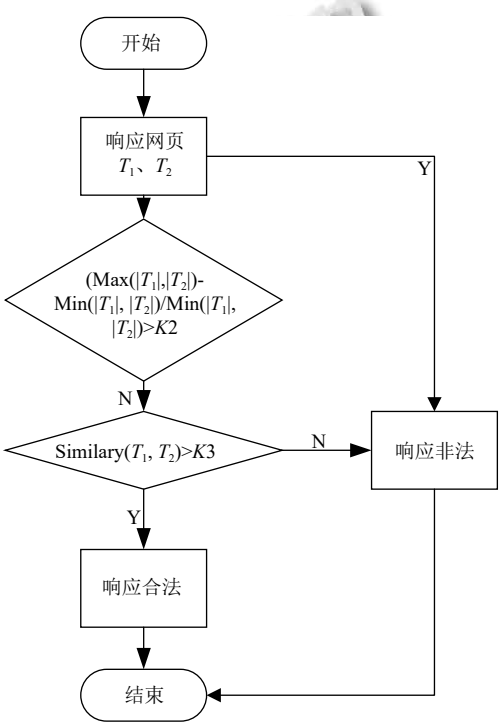


图 4 表决器处理流程图

4 实验结果分析

拟态 Web 服务器网页防篡改应用场景中, 计算效率和计算准确性是两个重要的评价指标. 本文中, 为验证本方法的可用性, 采用计算效率和计算准确性两个指标与现使用算法进行比较. 实验将中电集团某研究所的官方网站部署到拟态 Web 服务器上. 基于字符串编辑距离的计算方法是通过计算两个字符串的编辑距离判断相似性, 于是在现有经典算法中将响应网页看成一个字符串进行完全比较. 改进简单树匹配方法中, 把响应网页所转换成的 DOM 树进行匹配. 首先, 分别对具有差异性的 8 对网页利用两种算法计算相同请求中异构执行体响应网页的相似度, 记录相似度值和计算所用时间.

实验中, 保存了 Ubuntu 和 Centos 两个虚拟机执行体 Web 服务器中有差异的 8 对网页, 在表决器中分别使用经典方法和改进简单树匹配方法计算每对网页之间的相似度值并分别记录耗时. 测试环境为 CPU: E5 4 核; 内存: 8 GB; 操作系统: CentOS-7 64 位.

分别设计基于经典算法和本文算法的表决器, 对 8 对网页相似度进行计算. 图 5 和图 6 为得到的相似度计算结果. 图中结果显示, 两种算法所计算的正常网页的相似度的结果差异不明显, 改进的算法对网页差异容忍度比经典算法略高, 但是差异不大, 不会对比较结果造成明显影响.

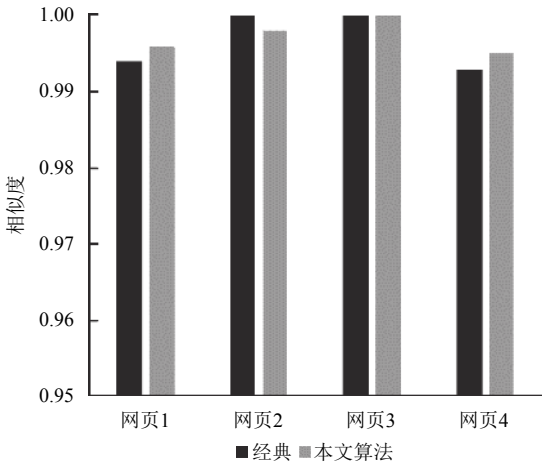


图 5 a 网站相似度求解算法结果对比

表 1 中记录 8 对网页分别采用经典方法和本文改进简单树匹配算法计算相似度的耗时结果, 对比表中数据可以看出, 本文所用改进算法大幅降低了计算耗时, 原因是改进算法中, 仅对网页可展示部分以及部分脚本进行比较计算, 大大缩减了需要计算的字符串量. 从表中还可以看出, 本文所用改进方法在计算网页的 DOM 树相似性时, 计算耗时与编辑距离和节点距离并不是线性关系. 其原因是, 改进的字符串匹配算法在比较网页的相似度时, 采用的是递归的方式遍历整棵 DOM 树, 网页被篡改的位置越靠近根节点, 所需计算时间越短, 差异地方越靠近叶节点, 所需时间越长. 实际应用场景中, DOM 树叶节点对应着网页页面上重要性相对低的位置, 这些位置被篡改价值低, 通常这些位置不会发生篡改, 因此改进方法可以防范常规的网页篡改攻击.

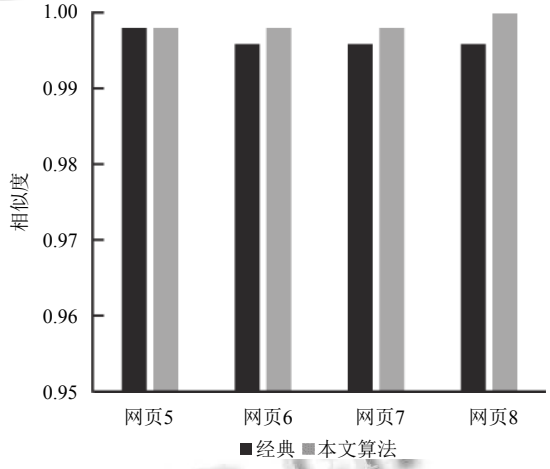


图 6 b 网站相似度求解算法结果对比

表 1 网页相似性计算时间

	网页大小 (KB)	编辑距离	节点距离	时间 (s)	
				经典算法	改进算法
网页 1	14	3	1	51.77	5.20
网页 2	21	8	2	202.27	121.52
网页 3	25	21	3	267.73	163.13
网页 4	36	25	2	306.05	10.19
网页 5	18	9	2	132.57	71.09
网页 6	29	7	5	146.14	83.17
网页 7	23	13	4	164.94	106.47
网页 8	19	9	4	225.67	117.77

实验 2, 在拟态防御系统中, 针对 Centos 虚拟机的在线 Web 服务器发起篡改网页攻击, 改变本实验中网页 4 的信息. 篡改形式包括更改官网标题、篡改官网超链接信息以及在网页上嵌入恶意脚本信息等. 分别利用改进简单树匹配方法和现有经典方法计算被篡改

网页的相似度. 根据网页被篡改前后相似度的变化程度判断算法性能, 理论上, 变化幅度越明显, 越能反应网页被篡改的实际情况.

从图 7 中可看出, 网页被篡改后利用经典算法和改进简单树匹配方法所计算的相似度均出现一定程度下降. 但从图中曲线变化趋势来看, 针对前两种篡改手段, 改进简单树匹配算法在网页被篡改后有较明显的下降趋势, 在网页嵌入恶意脚本攻击情况下, 也保持了和现有经典方法相近的趋势. 实验结果表明, 在拟态 Web 服务器中, 与现使用方法相比, 本文所采用改进简单树匹配算法能够在一定程度适应异构执行体 Web 服务器自身差异的基础上, 提高了拟态防御系统中表决器对于网页相似度计算所要求的准确性和计算效率.

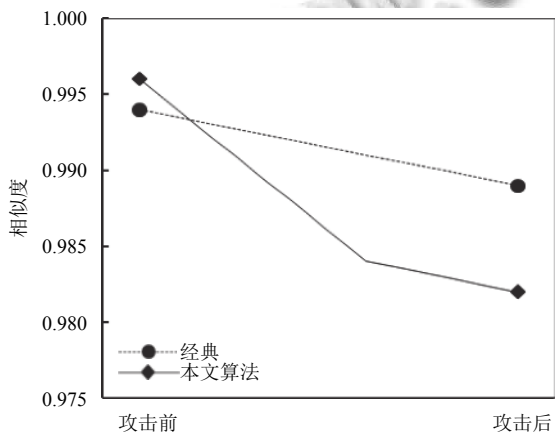


图 7 篡改网页检测效果

5 结论与展望

针对拟态 Web 服务器的应用场景, 结合字符串编辑距离计算方法和简单树匹配算法, 本文设计了一种符合拟态 Web 服务器系统中表决器需求的改进简单树匹配算法, 并用其计算拟态 Web 服务器中异构执行体响应网页的相似度. 实验结果表明, 本文所使用的算法更适用于拟态 Web 服务器异构环境下的表决器判决场景, 在测试环境中提高了表决器的计算效率和准确性, 对于被篡改网页有明显检测效果. 今后将对插入脚本篡改攻击检测不明显、深层节点篡改检测效率优化等方面做进一步的研究.

参考文献

1 邬江兴. 拟态计算与拟态安全防御的原意和愿景. 电信科学, 2014, 30(7): 2–7. [doi: 10.3969/j.issn.1000-0801.2014.07.001]
2 仝青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现. 软件学报, 2017, 28(4): 883–897. [doi: 10.13328/j.cnki.jos.005192]
3 马博林, 张铮, 刘健雄. 应用于动态异构 web 服务器的相似度求解方法. 计算机工程与设计, 2018, 39(1): 282–287.
4 姜华, 韩安琪, 王美佳, 等. 基于改进编辑距离的字符串相似度求解算法. 计算机工程, 2014, 40(1): 222–227. [doi: 10.3969/j.issn.1000-3428.2014.01.047]
5 祁钰, 关毅, 吕新波, 等. 网页结构树相似度计算. 黑龙江大学自然科学学报, 2009, 26(5): 627–632. [doi: 10.3969/j.issn.1001-7011.2009.05.012]
6 张瑞雪. 基于 DOM 树的网页相似度研究与应用[硕士学位论文]. 大连: 大连理工大学, 2011.
7 何昕, 谢志鹏. 基于简单树匹配算法的 Web 页面结构相似性度量. 第二十四届中国数据库学术会议论文集 (研究报告篇). 海口, 中国. 2007. 1–6.
8 陈秋. 移动互联网内容相似性研究[硕士学位论文]. 武汉: 华中科技大学, 2013.
9 林森杰, 刘勤让, 王孝龙. 面向拟态防御系统的竞赛式仲裁模型. 计算机工程, 2018, 44(4): 193–198.
10 斯雪明, 王伟, 曾俊杰, 等. 拟态防御基础理论研究综述. 中国工程科学, 2016, 18(6): 62–68.
11 郑小昌. 基于可信度和语义相似度的网页信息甄选研究[硕士学位论文]. 南京: 南京理工大学, 2016.
12 黄亮, 赵泽茂, 梁兴开. 基于编辑距离的 Web 数据挖掘. 计算机应用, 2012, 32(6): 1662–1665.