改进的AC-BM字符串匹配算法

万国根，秦志光

(电子科技大学计算机科学与工程学院 成都 610064)

【摘要】提出了改进的AC-BM算法，将待匹配的字符串集合转换为一个类似于Aho-Corasick算法的树状有限状态自动机。匹配时，采取自后向前的方法，并借用BM算法的坏字符跳转和好前缀跳转技术。改进的AC-BM算法借助BMH算法思想，取消了原AC-BM算法的好前缀跳转，并对坏字符跳转部分的计算进行优化。新算法修改了skip的计算方法，不再保留每个节点的好前缀跳转参数及坏字符跳转参数，因此匹配只与当前匹配字符有关，而与当前节点无关，可以实现大小写正文的识别。

关 键 词 算法； 字符串匹配； 内容分析； 入侵检测

中图分类号 TP393.08 文献标识码 A

Improved AC-BM Algorithm for Matching Multiple Strings

WAN Guo-gen，QIN Zhi-guang

(School of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract ACBM is a Boyer-Moore like algorithm applied to a set of keywords held in an Aho-Corassick like keyword tree that overlays common prefixes of the keywords. The algorithm takes the best characteristics of both the Boyer-Moore and Aho-Corasick. Based on the idea of Boyer-Moore-Horspool, we make an improvement to AC-BM algorithm. In the improved version, the Good Prefix Shift is not performed, the Bad Character Shift function is improved, the Goto procedure is also modified which do not keep the parameters of Good Prefix Shift and Bad Character Shift.

Key words algorithm; string matching; content analysis; intrusion detection

字符串匹配是指在文本$\text{Text}_t=t_1t_2\cdots t_n$中检索子串$\text{Pat}_p=p_1p_2\cdots p_m$的所有出现。字符串匹配可分为单字符串匹配和字符串集合匹配两种。单字符串匹配算法最著名的是KMP(Knuth-Morris-Pratt)算法和BM(Boyer-Moore)算法。KMP算法实现了无回溯匹配，字符串中的每个字符只匹配一次，时间复杂度为$O(n+m)$[1]；BM算法采用跳跃方式，匹配时跳过不需匹配的字符，最优情况下的时间复杂度为$O(n/m)$，平均情况下也大大优于KMP算法[2]；文献[3]取掉BM算法的好后缀试探(Good Suffix Heuristic)，形成BMH(Boyer-Moore-Horspool)算法，实验证明BMH算法性能在自然语言环境下比原始BM算法还要好。

字符串集合匹配是从文本Text中一次查找多个字符串$P_1, P_2, \cdots, P_m$的所有出现，最经典的是AC(Aho and Corasick)算法。该算法将待匹配的多个字符串转换为树状有限状态自动机，然后进行扫描匹配，最优情况和平均情况的时间复杂度都为$O(n)$[4]；文献[5]提出了一种类似于AC和BM的算法，即Aho-Corasick_Boyer-Moore(AC_BM)混合算法。该算法根据包过滤的规则前缀共同信息较多的特点，采取自后向前的搜索方法，因此在速度方面具有较高的优越性[6]。

本文对AC-BM算法提出改进，并根据BMH算法的思想，取消好前缀跳转(Good Prefix Shift)，测试结果表明，该算法在数据包过滤、内容分析与审计等应用中，优于AC-BM算法。

1 AC-BM算法描述

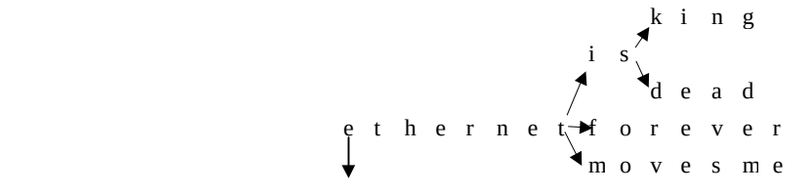
AC-BM算法将待匹配的字符串集合转换为一个类似于Aho-Corasick算法的树状有限状态自动机，但构建时不是基于字符串的后缀而是前缀。匹配时，采取自后向前的方法，并借用BM算法的坏字符跳转(Bad

收稿日期：2003－12－12
基金项目：国家信息产业部资助项目(2001–研–0–024)
作者简介：万国根(1963－)，男，在职博士生，高级工程师，主要从事网络安全理论与技术、中文信息处理方面的研究.

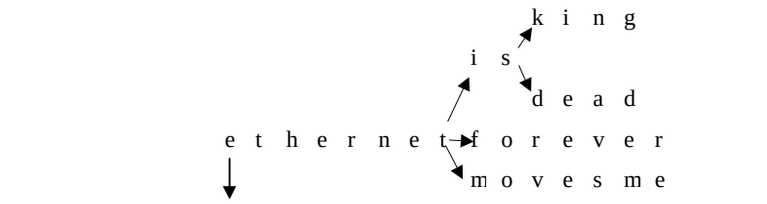
Character Shift)和好前缀跳转(Good Prefix Shift)技术。

坏字符跳转即当字符串树中的字符与被匹配内容*x*失配时，将字符串树跳转到下一个*x*的出现位置，如果*x*的字符串树不存在，则将字符串树向左移动字符串树的最小字符串长度。好前缀跳转即当字符串树中的字符与被匹配内容*x*失配时，将字符串树跳转到字符串树中一个与被测正文部分等同的位置。这个等同部分可以是字符串树中某字符串的子串(子串跳转)，也可以是一个字符串的后缀(后缀跳转)。

例如，设有规则ethernetmovesme，ethernetisking，ethernetisdead和ethernetforever，要检查的数据包内容为nothingtoworryaboutinthis。首先，构造字符串有限状态自动机，如图1a。匹配从字符r开始,显然r与e不匹配，由于字符串树中下一个r出现在深度5的位置，字符串树的最小字符串长度为14，根据坏字符跳转，字符串树可以安全向左移动5个字符，如图1b。



a. 移动前



b. 移动后

图1 坏字符跳转

当既有好后缀跳转，又有坏字符跳转时，则判断如果有后缀跳转，就使用好前缀跳转(子串跳转与后缀跳转的最小值)；否则，使用好前缀跳转与坏字符跳转的最大值。

2 AC-BM算法的改进

AC-BM算法的跳转过程基于BM算法。然而，正如BMH算法所证明的，简单搜索算法常常比每次跳转更多字符，但每次跳转需花费更多代价的算法优越。Horspool仅仅使用BM算法的坏字符跳转实现BMH算法，但BMH算法的实际性能并不比BM算法差；相反，由于减少了好后缀跳转的比较步骤，在某些情况下性能比BM算法还要优越。因此，可以根据Horspool的思想，简化AC-BM算法的复杂性，并对坏字符跳进行改进，形成一种新的算法。以下是改进的AC-BM算法描述。

输入：字符串树root，正文$T[1..n]$，minlen_l，maxlen_l；

输出：字符串位置。

算法：

```
i=n–minlen_l;
while(i>=0)
{
j=i;
currentNode=root;
while(currentNode->child[T[j]]!= NULL)
{
    currentNode=currentNode->child[T[j]]);
    if((currentNode->label)>=0)
```

```
        {
            output(currentNode)
        j++;
            if(j> n)break;
        }
    }
    i=i-skip(T [i]);
    }
```

其中skip的计算公式为:

if 存在k，j使$P_k[j]$=char，$0<j\leqslant \text{minlen}_l$

skip(char)=min{j |$P_k[j]$=char，$0<j\leqslant \text{minlen}_l$，$1\leqslant k\leqslant q$，$q$为字符串树中字符串的个数;

else

skip(char)=minlen_l

AC-BM算法的改进主要体现在：

(1) 修改了skip的计算方法，使其只与当前匹配字符有关，而与当前节点无关。

(2) 取消了好前缀跳转的计算。

(3) AC-BM算法在扫描正文前需要将正文转换为小写，而这在有些应用中是不切实际的。改进之后，由于不再保留每个节点好前缀跳转参数及坏字符跳转参数，因此，新算法可以实现大小写正文的识别。

改进后的AC-BM算法与原AC-BM算法具有相同的最优时间复杂度。设字符串集合中，字符串最小长度为minlen_l，最大长度为maxlen_l,待匹配正文长度为n，则在最优情况下，时间复杂度为$O(n/\text{minlen}_l)$,在最坏情况下，时间复杂度为$O(n*\text{maxlen}_l)$。但在平均情况下，由于取消好前缀跳转的判断，因此，特别是字符集较大(如正文中包括中文、日文、英文等)、字符串长度较短时，skip大的概率就增加[7]，改进后的AC-BM算法比原AC-BM算法具有更好的性能。

3　测试结果

为测试改进后的AC-BM算法的性能，在Windows2000下使用VC6.0编译过的snort2.0作为测试本算法的入侵检测系统，使用规则为snort2.0默认值；测试环境为CPU Intel P4，2.0G，内存512M,操作系统为Windows 2000 Server。

测试数据采用1999年美国国防部高级研究计划局(DARPA)做IDS评估时使用的数据集。数据集取自一个虚拟的美国空军基地的7个星期的网络流量，其中包含大量的正常网络流量和各种攻击，具有很强的代表性。以下是第一个星期的数据(训练数据)。

表1　测试结果

	测试数据	AC-BM算法/s	改进的AC-BM算法/s
星期一	inside.tcpdump(325 M)	208.996 7	200.006 2
	outside.tcpdump(308 M)	196.190 8	185.166 9
星期二	inside.tcpdump(325 M)	141.048 6	133.096 4
	outside.tcpdump(310 M)	131.440 4	124.099 4
星期三	inside.tcpdump(367 M)	285.567 6	281.010 3
	outside.tcpdump(351 M)	272.888 7	270.385 6
星期四	inside.tcpdump(527 M)	219.875	214.416 6
	outside.tcpdump(493 M)	203.215 6	218.437 6
星期五	inside.tcpdump(294 M)	144.34 43	139.381 6
	outside.tcpdump(271 M)	131.062 3	125.129 6

(下转第541页)