入侵检测系统中BM算法的改进

薛传庆[1],韩明畅[2],金伟信[2]
(1.解放军理工大学 通信工程学院,江苏 南京 210007;
2.中国电子系统设备工程公司安全保密中心,北京 100039)

摘 要:随着网络安全问题的日益突出,入侵检测技术也成为当前研究的热点,模式匹配算法是入侵检测系统(IDS)中一种重要算法,直接影响到系统的准确性和实时性。在研究BM算法和分析现有改进算法的基础上,提出了一种新的改进算法。该算法利用了末字符和下一个字符在模式串中首次出现的位置、存在性、唯一性的判断来增加模式串移动距离,利用记录因子记录上次匹配过程中的匹配后缀来减少比较次数,从而有效地加快模式匹配的速度,提高入侵检测的效率。

关键字:入侵检测;模式匹配;BM算法;模式串
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2011)06-0136-04

Improvement of BM Algorithm in Intrusion Detection System

XUE Chuan-qing[1],HAN Ming-chang[2],JIN Wei-xin[2]
(1. Institute of Communication Engineering,People's Liberation Army University of Technology,Nanjing 210007, China;
2. Security Research Center, China Electronic Equipment System Engineering Company, Beijing 100039,China)

Abstract:With the online security especially the invading detection technique is becoming a hot topic. Pattern matching algorithm is an important method in intrusion detection system (IDS) which directly influences the accuracy and real-time performance of the system. In this paper, on base of researching BM algorithm and analyzing other improved algorithms, a new improvement algorithm is proposed. The algorithm can increase the shift distance by making use of the first position, judged the existence and the uniqueness of the last character and next character of the pattern string, and it can decrease the comparison times by the memory a factor the last match record in the process of matching the suffix. In this way, the new algorithm improves the match speed and the efficiency of the intrusion detection system.

Key words:invading detection; pattern matching; BM algorithm; pattern string

0 引 言

随着网络的快速发展,其安全问题也日益凸现出来,然而计算机安全和网络安全的最大威胁就是网络入侵。为了抵御这些威胁,将它们的危害降低到最低限度,入侵检测技术作为一种积极主动的安全防护技术成为当前的研究重点和热点[1]。IDS[2,3]在网络信息安全中扮演着重要角色,它常采用特征字符串描述入侵行为,通过对网络数据包进行字符串匹配来判断入侵行为。在实际的网络运行中,IDS的性能大大依赖于精确字符串匹配算法的速度,因此寻找更有效的字符串匹配算法是当务之急。

收稿日期:2010-11-25;**修回日期**:2011-02-14
基金项目:总装备部科研计划项目(20091315)
作者简介:薛传庆(1983-),男,江苏淮安人,硕士,研究方向为网络安全;金伟信,高级工程师,硕士,研究方向为网络安全。

目前,IDS大多采用单模式匹配算法,单模式匹配算法有很多,文中着重分析BM算法及其相关改进算法,结合其优点对BM算法作出新的改进。

1 相关算法分析

模式匹配描述如下:对于给定的文本 $T=T_1,\cdots,T_n$ 和模式串 $P=P_1,\cdots,P_m$,其中 $n>>m$,如果能在文本串 T 中寻找到等于模式串 P 的子串,称其匹配成功[4],否则称其匹配失败。

1.1 BM算法

1977年,Boyer和Moore提出了BM算法[5]。BM算法采用两种启发性规则——Badchar(坏字符)和Goodsuffix(好后缀)规则[6],分别计算模式串在两种规则下的移动值,然后取两者中较大者,尽可能地向右移动模式串P[7],直至匹配成功。

(1)坏字符规则。

用函数 Badchar(c)来计算移动量,其中 c 为失配时 T 中相应位置上的字符,Badchar(c)可描述如下:

Badchar(c) =

$$\begin{cases} m; c \text{ not in } P \text{ or } c = P_m \text{ 且 } c \neq P_j (1 \leq j \leq m-1) \\ m-j,\ j = \max\{j \mid P_j = c, 1 \leq j \leq m-1\}; \text{others} \end{cases}$$

当 T 在 c 处与 P 失配时,则下次应移动 dist(c) − ($m-j$) 远的距离,其中 j 为 c 在本次匹配时其在模式串中的位置。

(2) 好后缀规则[8]。

从 P 的最右端开始寻找前导字符不是 $P[j]$ 的字符串 u(即 $T[i+j+1\cdots i+m-1]=P[j+1\cdots m-1]$),如果存在该子串和 u 对齐;如果不存在这样一个子串(即在 P 中只有一个后缀 u),则在 u 的后缀中查找与 P 的前缀相同的子串 v($|v| \leq |u|$),并让这个前缀 v 尽可能大,这样可以加快匹配速度。BM 算法的匹配过程见表1。

表1　BM 算法匹配过程

C	A	B	D	A	C	A	B	C	B	C	B	C	D	B	D	B	A	B	C	D	A	C	B	C
C	A	B	C	B	C	B	C																	
	C	A	B	C	B	C	B	C																
			C	A	B	C	B	C	B	C														
								C	A	B	C	B	C	B	C									
														C	A	B	C	B	C	B	C			

1.2 BMLT 算法

BMLT(BM-Less Time)算法[9]针对部分匹配的结果设置了记录因子,前提是当上次匹配中采用了 Good_Suffix 移动时,即出现了 P 和 T 部分匹配的情况。若本次匹配中所记录的部分匹配子串长度小于上次的匹配长度,则移动距离为 mini_shift。此时记录因子用 u 来表示,本次匹配后缀用 v 来表示,P 的一个后缀子串是 uzv。设 e 和 f 分别是 P 和 T 中导致此次失配的字符,则 ev 就是 P 的一个后缀。如图2所示[10],由于 $|u|>|v|$,所以 ev 也是 u 的后缀。令字符 e 和 f 在 T 中的距离为 x,由于 u 是 uzv 的边界,则 P 的后缀 uzv 就包含一个字符区间,其长度为 $x=|zv|$。由于 $|u|<x$,再考虑 $e \neq f$,则 u 中不可能包含字符 f,此时应至少把子串 u 移至 f 的右侧,即最小移动量 mini_shift = $|u|-|v|$。

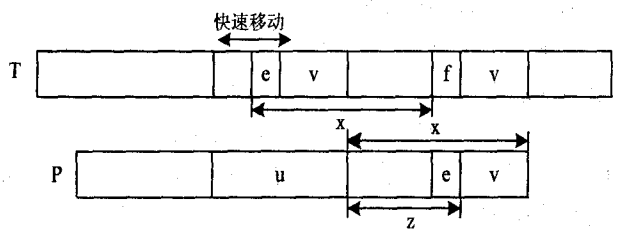

图2　当 $|u|>|v|$ 时启动 mini_shift

当 $|u|>|v|$ 时,若 Bad_char 移动值大于 Good_ suffix 移动值和 mini_shift,则实际移动距离肯定不小于 $|u|+1$。若 P 的移动距离 $p \in$ (mini_ shift, $|u+1|$),此时会使 P 中字符 c 和 d 对应 T 中的字符相同,而由于 $c \neq f$,所以匹配失败。因此,此时实际移动距离必须不小于 $|u|+1$,即:shift = max($|u|+1$, Bad_char)。如图3所示,移动过程如表2所示。

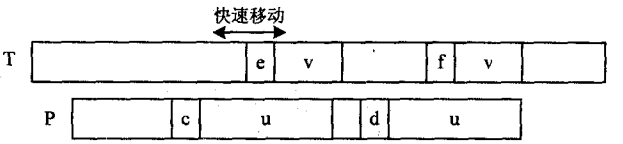

图3　$|u|<|v|$ 时

表2　BMLT 算法匹配过程

C	A	B	D	A	C	A	B	C	B	C	B	C	D	B	D	B	A	B	C	D	A	C	B	C
C	A	B	C	B	C	B	C																	
	C	A	B	C	B	C	B	C																
			C	A	B	C	B	C	B	C														
								C	A	B	C	B	C	B	C									
														C	A	B	C	B	C	B	C			

1.3 QS 算法

QS(Quick Search)算法[11]是由 D. M. Sunday 在1990年提出的一种高效的匹配算法。在对 $T[j\cdots j+m-1]$ 的一次匹配后,跳转的距离至少为1,因此,字符 $T[j+m]$ 必然在下一次匹配中。所以,可以由字符 $T[j+m]$ 查阅不良字符跳转表来决定跳转距离。假设当前匹配窗口为 j,不良字符跳转表依据当前匹配窗口的下一个字符来进行跳跃,$x=T[j+m]$,则跳跃方式分为两种[12]:

(1)若 P 中含有字符 x,则在 P 中找出字符 x 的最右出现位置 i,则跳跃距离为 $m-i$。

(2)若 P 中不含 x,则 P 的跳跃距离为 $m+1$。

表3　QS 算法匹配过程

C	A	B	D	A	C	A	B	C	B	C	B	C	D	B	D	B	A	B	C	D	A	C	B	C
C	A	B	C	B	C	B	C																	
	C	A	B	C	B	C	B	C																
			C	A	B	C	B	C	B	C														
								C	A	B	C	B	C	B	C									
														C	A	B	C	B	C	B	C			
															C	A	B	C	B	C	B	C		
																C	A	B	C	B	C	B	C	

QS 算法只用了不良字符跳转表,它实质上是 BM 算法的一种简化,其主要特征是[13]:考察紧邻在当前匹配窗口的下一个字符,利用文本串中未出现在模式串里的字符来加快匹配速度。在实际应用中,QS 适用于大字符集而模式串较为短的情况。QS 算法匹配过程如表3。

2　改进后的算法

改进后算法从两个方面考虑:(1)从不良字符跳

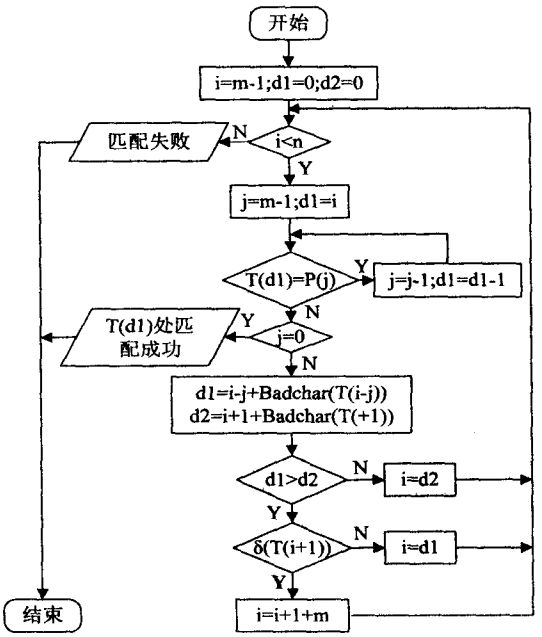

图 4 改进后算法流程(1)

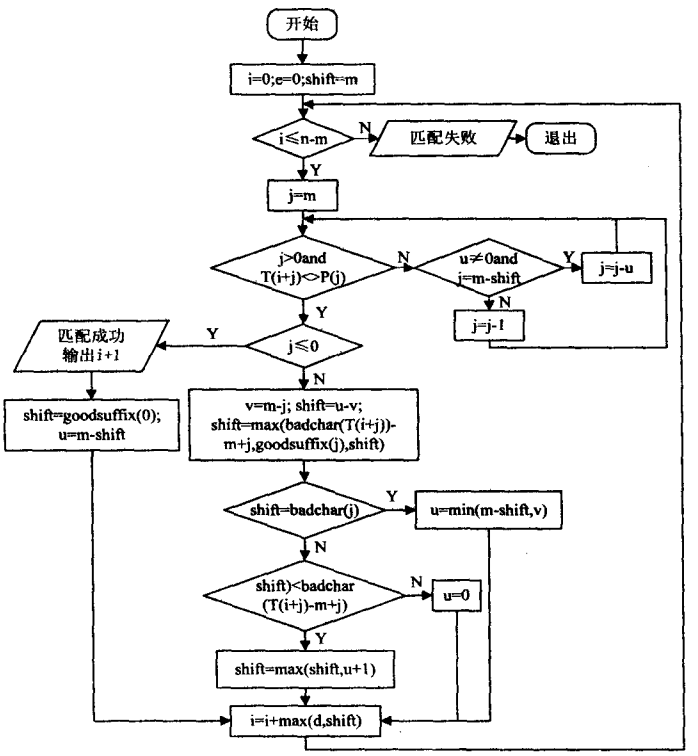

图 5 改进后算法流程(2)

转表方面来考虑,不良字符跳转表根据当前匹配窗口匹配失败的字符和当前窗口的下一个字符来进行跳转,跳转的距离是两者中较大的那个,算法流程如图 4 所示;(2)从好后缀方面来考虑,对部分匹配的结果设置记录因子,根据记录因子来进行跳转,跳转的距离是本次跳转与(1)中大者进行比较较大的那个,算法流程如图 5 所示。

改进后算法匹配过程如表 4 所示。具体过程如下:

表 4 改进后算法匹配过程

```
C A B D A C A B C B C B C D B D B A B C D A C B C
C A B C B C B C
  C A B C B C B C
        C A B C B C B C
                    C A B C B C B C
                                C A B C B C B C
```

(1)由当前匹配失败的字符 $T[i-j]$ 计算出右移距离 Badchar($T[i-j]$)和模式串末端对应的文本串字符 $T[i]$ 后一位字符 $T[i+1]$ 计算出 Badchar($T[i+1]$),移动后模式串末端对应的字符下标分别是 $d_1 = i+$ Badchar(T[$i-j$]), $d_2 = i+1+$ Badchar(T[$i+1$]),再比较 d_1 和 d_2 的大小,如果 $d_1 < d_2$,则将模式串末端移至 $T[\ d_2]$ 处进行下一轮匹配;如果 $d_1 < d_2$,则判断 $T[i+1]$ 在模式串中出现的次数,如果不出现或只出现一次,则直接右移 $m+1$ 个字符,如果出现多于一次,则将模式串末端右移至 $T[\ d_1]$ 处,再进行下一轮匹配。这里用 δ 函数判断 $T[i+1]$ 在模式串中出现的次数,δ 函数定义如下:

$$\delta(c) = \begin{cases} 1; c \text{ not in } P \text{ or } c \text{ in } P \text{ only once} \\ 0; c \text{ in } P \text{ more then once} \end{cases}$$

(2)在此次匹配过程中,当比较到模式串 P 中已记录的上次匹配后缀时,在计算下次移动量时候需要考虑一个最小移动量 shift。移动后模式串末端对应的字符下标是 $d_3 = i +$ shift ,然后将 d_3 与 d_1 和 d_2 中较大的 d 进行比较,取最大者。

给定文本串:CABDACABCCBCDBDBABCD

ACBC 和模式串:CABCBCBC,文中分别用 BM、BMLT 和 QS 算法以及改进后的算法进行匹配,匹配次数分别为 5、5、7 和 5 次,比较次数分别为 19、15、25 和 12 次。一次最大移动量分别为:7、7、9 和 9 个字符,可见该改进算法虽然一次最大移动量相差不大,但其比较次数较之其他三种算法均有很大的进步。

可见,影响算法效率最主要的因素是产生最大移动量的概率和窗口的比较次数。BM 和 BMLT 算法产生最大位移的情况为与模式串末位对齐的那个文本串字符在模式串中未出现;而 QS 算法产生最大位移的情况为与模式串对齐的文本串字符的下一个字符在模式串中未出现;在本文的算法中产生最大位移量的情况为下一个字符在模式串中未出现、末字在模式串中未出现或者末字符下一个字符在模式串中未出现。所以本文的算法产生最大移动量的概率比 BM、BMLT 和 QS 算法大,比较次数少。由于 BM 算法和 QS 算法都没有使用记录因子,而本文使用了记录因子具有记录功能。所以综合来讲本文的算法具有一定的优越性。

3 实验结果分析

选择一篇英文文学作品作为测试语料,大小为4.83MB,选择其中长度为5-10个英文单词、短句作为模式串,总共选择5个,统计各模式串在测试语料中出现的总次数,占用CPU的时间,总的尝试次数和总的比较次数。实验环境为:CPU为Pentium(R),2.8 GHz,内存为1GHz,操作系统为Windows XP,编译环境为VC++ 6.0。实验数据如表5所示。

表5　算法比较

算法	CPU时间/ms	尝试次数	字符比较次数
BM算法	7182	265432828	272628876
BMLT算法	7378	259779857	264857429
QS算法	7424	251875392	256438952
改进算法	5612	219847372	220775265

从实验结果来看,改进算法较算法在占用时间、尝试次数和比较次数上效率都有不小的提高,其中占用CPU时间是BM算法的78.14%,总的尝试次数是BM算法的82.83%,总的字符比较次数是BM算法80.98%,而其他两种算法显然也没有本算法效率高。

4 结束语

对BM算法和现有的BM改进算法进行了简要的介绍,结合其各自的优点提出了一种新的BM改进算法。从理论分析和实验结果来看,改进后的BM算法的匹配次数和比较次数有所减少,匹配时间也缩短了。若将其应用到入侵检测系统的检测引擎中,可以提高系统检测的速度,改善系统性能。

文中主要从以下两个方面进行对BM算法优化:

(1)计算下次移动量时要同时考虑本次匹配的坏字符和本次匹配时模式串对应文本串后一个字符,比较这两个字符产生的跳转距离,选择较大的那个,从而扩大了一次最大移动量;

(2)在匹配过程中,纪录因子时,可能跳过此记录因子的一部分或全部,从而实现跳跃式比较,进而会减小此次匹配过程中的字符比较次数实现快速匹配。

参考文献:

[1] Desai N. Increasing Performance in High Speed NIDS[EB/OL]. 2002-03-15. http://www.linuxsecurity.com.

[2] 王永全.入侵检测系统(IDS)的研究现状和展望[J].通信技术,2008,41(11):139-143.

[3] Iheagwara C, Blyth A. Evaluation of the performance of ID systems in a switched and distributed environment: the RealSecure case study[J]. Computer Networks, 2002, 39(5): 93-112.

[4] 杨薇薇,廖　翔.一种改进的舰模式匹配算法[J].计算机应用,2006,26(2):318-319.

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

[6] 张红梅,范明钰.模式匹配BM算法改进[J].计算机应用研究,2009,26(9):3249-3252.

[7] 巫喜红,凌　捷.BM模式匹配算法剖析[J].计算机工程与设计,2007,28(1):29-31.

[8] 袁静波,郑吉森,丁顺利.一种BM模式匹配算法的改进[J].计算机技术与发展,2009,19(7):105-107.

[9] 杜　丰.入侵检测中BM模式匹配算法的研究和改进[D].杭州:浙江工业大学,2009.

[10] 顾　钧.基于网络入侵模式匹配的BM算法研究与优化[J].微计算机信息,2009,25:44-46.

[11] Daniel M S. A very fast substring search algorithm[J]. Communications of the ACM, 1990, 33(8):132-142.

[12] 章　张.基于层次分类的网络安全监管系统中串匹配算法的设计与实现[D].南京:南京理工大学,2004.

[13] 赵玲涛.基于内容的安全审计跟踪算法及其应用研究[D].上海:上海交通大学,2007.

(上接第135页)

-Round AES[C]//Fast Software Encryption 2008, Lecture Notes in Computer Science5086. [s.l.]:[s.n.],2008:116-126.

[4] Koblitz N. A course in number theory and cryptography[M]. New York:Springer-Verlag,1987.

[5] 韦宝典,刘东苏,王新梅.一种新Square攻击[J].西安电子科技大学学报(自然科学版),2003,30(4):473-476.

[6] 钟名富,胡予濮,陈　杰.分组加密算法SMS4的14轮Square攻击[J].西安电子科技大学学报(自然科学版),2008,35(1):105-109.

[7] Dunkelman O, Keller N, Shamir A. Improved Single-Key Attack on 8-round AES[EB/OL]. 2010. http://eprint.iacr.org/.

[8] Demirci H, Taskm I, Coban M, et al. Improved Meet-in-the-middle Attacks on AES[C]//Lecture Notes in Computer Science,5922. [s.l.]:[s.n.],2009.

[9] 王美一,唐学海,李　超,等.3D密码的Square攻击[J].电子与信息学报,2010,32(1):157-161.

[10] 李清玲,李　超.变种Camellia对Square攻击的安全性[J].应用科学学报,2006,24(5):485-490.

[11] 贺也平,吴文玲,卿斯汉.对于5轮Camellia密码的Square攻击[J].中国科学院研究生院学报,2001,18(2):177-180.

[12] 冯国登,吴文玲.分组密码的分析和设计[M].北京:清华大学出版社,2000.