

面向入侵检测系统的模式匹配算法研究

姜庆民, 吴宁, 刘伟华

(西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对入侵检测系统对基于攻击特征的网络数据包的检测效率低和丢包率高的问题, 在分析典型的模式匹配算法的基础上, 提出了一种 Boyer Moor Horspool Fast(BMHF)匹配算法. 引入一个新的判断函数 $Q(X)$ 指出字符 X 在模式串中出现的次数, 当出现次数为 1 时可以利用已匹配的信息加大移动距离, 同时利用文本串中不匹配字符后面的一个字符进行匹配, 从而得到一个移动距离. 将不同移动规则下获得的移动距离的最大值作为实际的移动距离, 依次进行, 直到匹配完成. 实验结果表明, BMHF 算法的 CPU 运算时间比典型的模式匹配算法可平均节省 5.7%, 平均匹配次数减少 12.5%.

关键词: 入侵检测; 模式匹配; 单次移动距离

中图分类号: TP312 **文献标志码:** A **文章编号:** 0253-987X(2009)02-0058-05

A Fast Pattern Matching Algorithm in Intrusion Detection System

JIANG Qingmin, WU Ning, LIU Weihua

(The School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A new fast pattern matching algorithm (called BMHF algorithm) is proposed to improve the detecting efficiency of network data based on attack signature and to decrease the packet loss rate in the intrusion detection system(IDS), by analyzing popular pattern matching algorithms such as BM algorithm, BMH algorithm and BMHS algorithm. A new function $Q(X)$ is employed in BMHF to identify the number of a character X matching in the pattern string. When $Q(X)$ equals one, the matched information is used to enlarge the shifting distance. A new shifting distance is also calculated according to the character next to non-matched characters in the pattern string. The maximum distance among these shifting distances is used in the BMHF algorithm. The process is repeated until the matching is completed. Experimental results show that the BMHF algorithm has better efficiency. The decrease in the operation time of CPU is 5.7 percent, and the decrease in the average number of matchings is 12.5 percent.

Keywords: intrusion detection; pattern matching; shifting distance

入侵检测是网络安全中的一个重要研究领域, 模式匹配则是入侵检测中基于攻击特征的网络数据包的检测技术, 也是入侵检测的关键技术之一. 在实际的入侵检测系统中, 模式匹配是检测过程中最费时的部分, 因此模式匹配算法的性能直接影响到整个入侵检测系统的效率. 基于此, 本文在对传统的模式匹配算法进行分析的基础上, 提出了一种基于 Boyer Moor Horspool Fast(BMHF)匹配算法, 以

此加快匹配速率, 提高入侵检测系统的效率.

1 相关算法分析

模式匹配是指, 在一个目标文本 T 中查找某个特定的子串, 使得这个子串与已知的模式串 P 相等. 如果在 T 中找到等于 P 的子串, 则称匹配成功, 否则称匹配失败. 模式匹配算法分单模式和多模式匹配算法. 传统的单模式匹配算法主要有 BF 算

法^[1-2]、KMP 算法^[3]、BM 算法^[4]及其改进算法。BF 算法实现简单,但匹配效率很低;KMP 算法比 BF 算法有所改进;BM 算法则比 KMP 算法的检测速度快 3~5 倍。在各种单模式匹配算法中,应用最广泛和最著名的是 BM 算法。

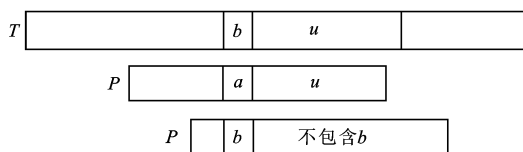
1.1 BM 算法

假设有长度为 n 的文本字符串 $T = T_1 T_2 \cdots T_n$ 和长度为 m ($m \leq n$) 的模式字符串 $P = P_1 P_2 \cdots P_m$, 当开始匹配时,文本字符串和模式字符串从左至右对齐,但匹配过程从右至左进行。当有不匹配的字符出现时,BM 算法根据 2 个偏移函数 Pdist 和 Goodsuffix 计算模式串向右移动的距离。

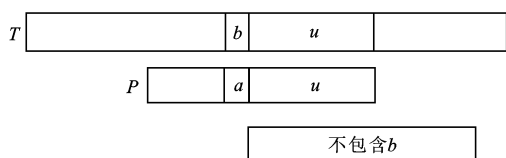
假设失配发生在 $T[i+j]=b$ 和 $P[i]=a$ 之处,Pdist、Goodsuffix 函数计算移动过程如下。

(1)Pdist 计算移动过程。该移动分为 2 种情况:

①将文本串字符 $T[i+j]$ 与它在模式串中从右到左首次出现的位置对齐,如图 1a 所示;②在模式串中如果没有 $T[i+j]$,则将模式串的最左端字符 ($P[1]$) 与 $T[i+j+1]$ 对齐,如图 1b 所示。



(a) $P[1 \cdots m-1]$ 中再次出现 $T[i+j]$

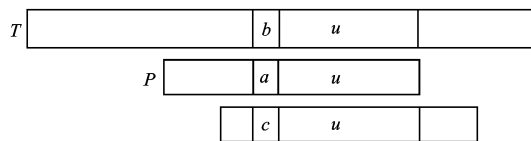


(b) $P[1 \cdots m-1]$ 中没有再次出现 $T[i+j]$

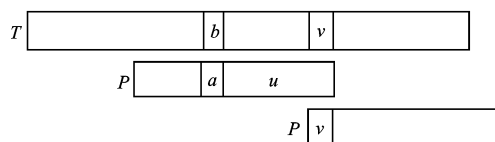
图1 Pdist 计算移动过程

(2)Goodsuffix 计算移动过程。该移动分为 2 种情况:①假设 u 是文本串与模式串比较过程中已经匹配了一个子串,如果 u 再次出现在模式串中,且 u 前面字符不是 a ,则将模式串最右端再次出现的且前面字符不是 a 的片段 u 与文本串 $T[i+j+1 \cdots m]$ 对齐,如图 2a 所示;②如果 u 未再次出现在模式中,或再次出现但 u 前面字符是 a ,将片段 $T[i+j+1 \cdots m-1]$ 中最长的后缀 v 与模式串前缀 v 对齐,若不存在这样的 v ,则模式串跳过 u ,如图 2b 所示。

每次偏移量取自 Pdist 和 Goodsuffix 计算出的较大的偏移距离,并将文本指针往右移,重新匹配,匹配成功则予以输出。



(a) u 出现在模式串中且前面字符不是 a



(b) u 在模式串中没有出现或出现且前面字符是 a

图2 Goodsuffix 计算移动过程

1.2 BMH 算法

BM 算法为了在不匹配的情况下确定最大的可行移动距离,利用了 2 种启发,即坏字符和好后缀移动,但是好后缀移动的预处理非常困难,且难以理解和实现。Horspool 于 1980 年提出了改进、简化的 BM 算法(BMH)算法^[5]。

BMH 算法的思想是:简单地使用文本中与模式最右端对齐的字符来决定向右的移动量,它仅利用了 Pdist 函数,即只用坏字符作移动启发。

1.3 BMHS 算法

文献[6]在 BMH 算法基础上提出 BMHS 算法,主要思想是:在计算 Pdist 函数时,利用不匹配字符的下一个字符 $T[k+1]$ 决定右移量。若下一个字符不出现在模式中,BMHS 算法的右移量比 BMH 算法大。通常情况下,BMHS 算法比 BMH 算法快,但 $T[k]$ 不出现在模式中,而 $T[k+1]$ 出现在模式中,即当 $Pdist[k] > Pdist[k+1]$ 时,BMHS 算法的效果不如 BMH 算法。

根据以上分析知,各种算法的目的都是为了使模式串和文本串在不匹配时将模式串尽可能向右移动最大的距离,但不同的是移动标准不一样。本文算法是在上述算法的基础上使移动距离更大,从而提高算法的效率。

2 BMHF 算法

2.1 BMH 算法匹配过程分析与改进

表 1 列出了 BMH 算法的匹配过程,并由此改进了 BMH 算法的匹配过程,描述如下。

(1)从最右端开始比较(即首先进行 $T[6]$ 与 $P[6]$ 的比较)。若匹配成功,则继续向左比较。 $T[3]$ 与 $P[3]$ 匹配失败,根据规则,将 $P[2]$ 与 $T[3]$ 对齐,从 $T[7]$ 和 $P[6]$ 重新开始进行第 2 次匹配。

表1 BMH 算法匹配过程

T 的长度	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T 的内容	a	p	e	r	c	h	e	t	r	c	h	e	a	r	c	h	
第1次匹配	h e a r c h																
第2次匹配	h e a r c h																
第3次匹配	h e a r c h																
第4次匹配	h e a r c h																
第5次匹配	h e a r c h																

注: apertchtrchearch 是文本串; hearch 是模式串。

(2) 直接将 $P[6]$ 与 $T[5+6]=T[11]$ 进行新一轮匹配。由于 $T[5]=P[5]="c"$ 已经是成功的匹配并且该字符在模式串 P 中仅出现过一次, 所以在第2次匹配过程中肯定还要将 $T[5]$ 与模式串 P 进行匹配。这里是 $T[5]$ 与 $P[4]$ 的匹配。因为模式串中字符“c”仅出现一次, 所以 $T[5] \neq P[4]$ 。因此, 第2次匹配不会成功, 也就是第2次匹配没有必要。

(3) 从 $T[11]$ 开始进行第3次匹配, 直到 $T[8]$ 与 $P[3]$ 匹配失败。字符“t”不在模式串 P 中, 根据规则, 应将 $P[1]$ 与 $T[9]$ 对齐, 从 $T[14]$ 和 $P[6]$ 重新开始进行第4次匹配。

(4) 从 $T[14] \neq P[6]$ 进行第4次匹配, 匹配失败, 再将 $T[14]$ 与 $P[3]$ 对齐进行第5次匹配。

(5) 第5次匹配成功。

根据上述改进思想, 可以发现改进后的匹配次数减少了1次。

2.2 BMHF 算法思想

我们通过对 BMH 算法匹配过程的分析与改进, 提出了一种新的快速模式匹配算法——BMHF 算法。

定义一个函数 $Q(X)^{[7-8]}$, 当字符 $X \in P$, 并且仅出现过1次时, $Q(X)$ 返回值为1, 否则返回值为0, 即

$$Q(X) = \begin{cases} 1 & \text{字符 } X \text{ 在模式串 } P \text{ 中仅出现过1次} \\ 0 & \text{字符 } X \text{ 在模式串 } P \text{ 中出现的次数大于1} \end{cases}$$

BMHF 算法的主要内容如下所述, 其流程如图3所示。

(1) 匹配自右向左进行, 如果匹配成功, 则使用 $Q(P[m])$ 检查 $P[m]$ 在模式串中出现的次数。如果只出现过1次, 则用一个变量 p 记录此次匹配过程中第1个 $Q(T[i])=1$ 的指针 i 的值, 继续进行匹配。

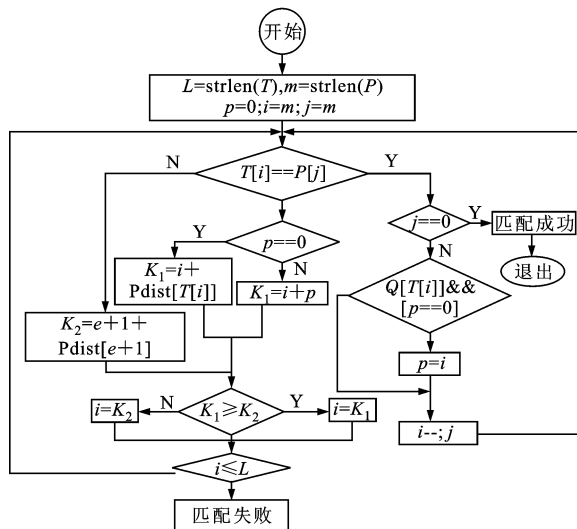


图3 BMHF 算法流程图

(2) 如果在某一处匹配失败且 $p \neq 0$, 表示已匹配字符中存在只出现过1次的字符, 则将文本串 T 的指针修改为 $i+p$, 并将其赋值为 K_1 ; 如果 $p=0$, 表示匹配失败处之后的字符在 p 中出现多次, 此时将 T 的指针修改为 $i+Pdist[T[i]]$ (假设在 i 处匹配失败), 并将其赋值为 K_1 。这相当于下一次匹配时文本串从 $T[Pdist[T[i]]]$ 开始, 模式串从 $P[m]$ 开始。

(3) 将由模式串末端对应文本串字符 $T[i]$ 下一位字符 $T[e]$ 得到的右移位置 $e+1+Pdist[e+1]$ 赋值给 K_2 。

(4) 判断 K_1, K_2 大小, 如果 $K_1 > K_2$, 将模式串末端 $P[m]$ 与 $T[K_1]$ 处对齐进行新一轮匹配, 同时 p 值置0; 如果 $K_2 > K_1$ 将模式串末端 $P[m]$ 与 $T[K_2]$ 处对齐。

(5) 新一轮匹配仍使用上面的规则。

函数 $Pdist$ 的描述如图4所示, 函数 $Q(x)$ 的描述如图5所示。

在文本串和模式串如同表1的情况下, BMHF 算法的匹配过程如表2所示。

表2 表1情况下 BMHF 算法的匹配过程

T 的长度	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T 的内容	a	p	e	r	c	h	e	t	r	c	h	e	a	r	c	h	
第1次匹配	h e a r c h																
第2次匹配	h e a r c h																
第3次匹配	h e a r c h																

由表2分析可得到如下结果。

```
int Pdist(char x,int d)
{
    int a=0,b,flag=0,m=d;
    while (a<m-1 && flag==0)
    {
        if (x!=P[a]) a++;
        else
        {
            flag=1;
            break;
        }
    }
    if (flag==0) return m;
    else
    {
        for (a=0;a<m-1;a++)
        {
            if (x==P[a])
            {
                b=a;
                b=m-b-1;
            }
        }
        return b;
    }
}
```

图 4 Pdist 函数描述

```
int Q (char x)
{
    int xx=0
    for (int t=0;t<P.size();t++)
    {
        if (x==P[t])
            ++xx;
    }
    if (xx==1)
        return 1;
    else
        return 0;
}
```

图 5 Q(x)函数描述

(1)在第 1 次匹配时, $i=3$ 不匹配, $Pdist[T[i]] = 4$. $T[5]=c$ 已匹配并且在 P 中只出现了 1 次,因此 $K_1=5+6=11$,而由 $T[11]$ 算出移动距离 $K_2=11$,则 $K_1=K_2$,所以 $P[6]$ 与 $T[11]$ 对齐进行第 2 次比较.

(2)在第 2 次匹配时, $i=8$ 不匹配, $Pdist[T[i]] = 6$. $T[10]=c$ 已匹配并且在 P 中只出现了 1 次,因此 $K_1=10+6=16$,而由 $T[12]$ 算出移动距离 $K_2=17$, $K_1<K_2$,所以 $P[6]$ 与 $T[11]$ 对齐进行第 3 次比较.

(3)第 3 次匹配成功.

表 3 不同算法匹配次数比较

测试文本	P	文件大小 /kB	匹配次数			
			BM 算法	BMH 算法	BMHS 算法	BMHF 算法
RFC1866	symbolically	135.0	11 976	12 081	11 135	10 237
网络数据库简介.doc	安装后需要重启浏览器	128.0	13 011	13 476	12 765	11 342
numeral.txt	12079238728	8.6	8 701	8 763	7 963	7 797

3 BMHF 算法的实验结果

评价一个模式匹配算法优劣程度的一个重要指标是字符的匹配次数和 CPU 的计算时间^[9-10]. 为此,本文为测试算法搭建了以下的实验环境:主机处理器是 Pentium IV 1.5 GHz,内存是 256 MB,操作系统采用 Windows XP SP2,编译器采用 MinGW Developer Studio,测试数据采用纯文本数据.

实验 1 分别用不同的文本和模式对 BM、BMH、BMHS、BMHF 算法进行匹配次数测试,测试结果如表 3 所示.

由表 3 结果可知,相对于 BM 算法,BMHF 算法在测试文本 1(RFC1866)中匹配次数减少了 14.5%,在测试文本 2(网络数据库简介.doc)中匹配次数减少了 12.8%,在测试文本 3(numeral.txt)中匹配次数减少了 10.3%,其平均匹配次数可以减少 12.5%. 相对于 BMH、BMHS 算法,BMHF 算法的匹配次数明显减少.

实验 2 选择一个 1 MB 的纯英语文本,分别用 P 的长度 $P'=5,10,20,30$ 的模式串对上述 4 种算法进行 CPU 运算时间测试,结果如表 4 所示.

由表 4 结果看出,对于相同的文本长度和模式串长度,不同算法执行匹配过程所需的 CPU 运算时间是不一样的. 相对于 BM 算法,当 $P'=5$ 时,BMHF 算法的 CPU 运算时间可以节省 4.6%;当 $P'=10$ 时,BMHF 算法的 CPU 运算时间可以节省 5.1%;当 $P'=20$ 时,BMHF 算法的 CPU 运算时间可以节省 5.9%;当 $P'=30$ 时,BMHF 算法的 CPU 运算时间可以节省 6.1%. BMHF 算法的 CPU 运算时间平均可以节省 5.7%. 相对于 BMH、BMHS 算法,BMHF 算法的 CPU 运算时间明显减少.

4 结束语

本文对传统的经典模式匹配算法进行了分析,并在此基础上提出了一种新的快速匹配算法——

表4 CPU 运算时间比较

算法	CPU 运算时间/ms			
	$P'=5$	$P'=10$	$P'=20$	$P'=30$
BM	16.732	13.547	11.431	9.472
BMH	16.254	13.278	11.292	9.315
BMHS	16.027	13.103	10.972	9.172
BMHF	15.962	12.851	10.751	8.893

BMHF 算法. 测试结果表明, BMHF 算法在文本串较长的情况下有较好的性能, 这符合入侵检测系统审计数据的要求, 有利于提高规则匹配的速度, 减少丢包率等.

参考文献:

[1] YAN Weimin, WU Weimin. Data structure [M]. Beijing, China: Tsinghua University Press, 1997.

[2] CHARRAS C, LECROQ T. Exact string matching algorithms [EB/OL]. [2008-02-05]. <http://www-igm.univ-mlv.fr/~lecroq/string>. 1997.

[3] KNUTH D E, MORRIS J H, PRATT V R. Fast pattern matching in strings [J]. SIAM Journal on Computing, 1977, 6(2):323-350.

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

[5] NIGEL-HORSPOOL R. Practical fast searching in

strings [J]. Software Practice and Experience, 1980, 10(6):501-506.

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

[7] 张娜, 张剑. 一个快速的字符串模式匹配改进算法 [J]. 微电子学与计算机, 2007, 24(4):102-105.

ZHANG Na, ZHANG Jian. A fast improved algorithm for pattern matching in string [J]. Microelectronics & Computer, 2007, 24(4):102-105.

[8] 李洋, 王康, 谢萍. BM 模式匹配改进算法 [J]. 计算机应用研究, 2004, 21(4): 58-59.

LI Yang, WANG Kang, XIE Ping. The improving algorithm of BM [J]. Application Research of Computers, 2004, 21(4):58-59.

[9] 牟永敏, 李美贵, 梁琦. 入侵检测系统中模式匹配算法的研究 [J]. 电子学报, 2006, 34(12A):2488-2490.

MU Yongmin, LI Meigui, LIANG Qi. The survey of the pattern matching algorithm in intrusion detection system [J]. Acta Electronica Sinica, 2006, 34(12A): 2488-2490.

[10] 蔡晓妍, 戴冠中, 杨黎斌. 一种快速的单模式匹配算法 [J]. 计算机应用研究, 2008, 25(1):45-49.

CAI Xiaoyan, DAI Guanzhong, YANG Libin. Faster algorithm for single pattern matching [J]. Application Research of Computers, 2008, 25(1):45-49.

(编辑 苗凌)

西安交通大学机械学院在高档数控机床科研领域获重大进展

近年来, 西安交通大学机械学院通过大力扶持青年教师, 积极引进高层次人才, 争取到高档数控机床研究领域多项国家重大科研攻关项目.

今年以来, 以卢秉恒院士为首席科学家的国家重点基础研究发展规划项目“超高速加工及其装备的基础研究”获得批准. 同时, 学院获得机床领域国家高技术研究发展计划项目 6 项, 其中重点项目 4 项, 分别是: 赵惠英教授负责的“大口径光学非球曲面加工机床和成套应用工艺”; 徐华教授负责的“高速主轴研究”; 姜歌东副教授负责的“高档数控机床综合动态特性在线测试系统”; 方素平教授负责的“复杂与大型齿轮测量系统”. 另外, 2006 年梅雪松教授在数控技术方面的研究成果也获得教育部科技进步一等奖.

从学科发展上来看, 高档数控机床是机械工程专业的核心研究方向之一, 也是衡量机械专业发展水平的重要标志. 随着学院机床领域的研究深入, 加之以上重点项目的实施推进, 机械学院在高档数控机床等重大装备领域的科研工作将会有更大的发展.

(来源: 交大新闻网 <http://xjtunews.xjtu.edu.cn>)