入侵检测系统中模式匹配算法的研究

牟永敏,李美贵,梁 琦
(北京信息工程学院计算机开放系统实验室,北京 100101)

摘 要: 互联网的开放性为信息共享和交互提供了极大的便利,但随之而来的网络安全问题也日益明显. 入侵检测作为一种主动的信息安全保障措施,有效的弥补了传统安全防护技术的缺陷. 主要分析了目前在入侵检测领域常用的模式匹配算法,如 KMP 算法和 BM 算法. 并在此基础之上,提出了一种新的模式匹配算法. 结果表明,改进后的算法具有更高的效率,有利于降低系统的丢包率.

关键词: KMP 算法; BM 算法; 入侵检测
中图分类号: TP393.08 **文献标识码**: A **文章编号**: 0372-2112 (2006) 12A-2488-03

The Survey of the Pattern Matching Algorithm in Intrusion Detection System

MU Yong-min, LI Mei-gui, LIANG Qi
(Open Computer System Laboratory, Beijing Information Technology Institute, Beijing 100101, China)

Abstract: The openness of Internet offers great convenience of information sharing and exchange, accompanied with crucial challenges to Information Security. As a kind of active measure of Information Assurance, Intrusion Detection acts as the effective complement to traditional protection techniques. On the base of analyzing KMP algorithm &BM algorithm, which are fashionable pattern matching algorithms in intrusion detection system at present, the author gives an new pattern matching algorithm in place of the BM algorithm. The results show that the improved pattern matching algorithm has better efficiency and is helpful to reduce systematical missing package rate.

Key words: KMP algorithm; BM algorithm; intrusion detection

1 引言

入侵检测技术是网络安全的一个重要领域. 网络级的入侵检测可以分为数据包的捕获、数据包的预处理以及对数据包进行攻击检测的过程. 模式匹配是入侵检测系统所使用的基于攻击特征的网络数据包检测技术,也是 IDS 中一个最基本、最关键的技术. 在实际的网络运行中,数据包的捕获速度与解释速度不能匹配,模式匹配速度的快慢直接影响到 IDS 的效率. 基于这个原因,本文中对传统的模式匹配算法进行了改进.

2 传统的模式匹配算法

2.1 KMP 算法[1,7]

主要思想:假设在模式匹配过程中,当前正执行到比较字符 t_i 和 p_j $(1 \le i \le n, 1 \le j \le m)$:

(1) 若 $t_i = p_j$,则继续向右匹配,即检查 t_{i+1} 和 p_{j+1} 是否匹配?

(2) 若 $t_i \neq p_j$,则考虑下列两种情况:

若 $j = 1$,则执行 t_{i+1} 和 p_1 的匹配检查,这相当于把模式、正文向右移动一个字符位置后再从头进行匹配.

若 $1 < j \le m$,则需要选择模式的某个适当的下标,记作 next[j],执行 t_i 和 $p_{next[j]}$ 的匹配. 此时,相当于把模式、正文向右移动 j-next[j] 个字符,模式中 next[j] 位置前面的各字符已与正文中 i 位置前的字符匹配,因此只需从模式的 next[j] 位置的字符开始继续匹配即可.

(3) 重复上述过程直到 $j > m$ 或者 $i > n - m + 1$ 为止.

KMP 算法的核心是构造 next 函数. 当 $1 \le j \le m$ 时,next[j] 定义如下:

$$\text{next}[j] = \begin{cases} \max\{k : 1 < k < j, \text{使得 } p[1 \ldots k-1] = p[j-(k-1) \ldots j-1]\} \\ 1, \text{对于所有的 } k : 1 < k < j, p[j-(k-1) \ldots j-1] \neq p[1 \ldots k-1] \end{cases}$$

2.2 BM 算法[2~4]

BM 算法的特点是考虑到在匹配比较过程中,不少情形是前面的许多字符都匹配而最后的若干个字符不匹配,这时若采取从左到右的方式扫描的话将浪费许多时间,因此,改为自右至左的方式扫描模式和正文,这样一旦发现当正文中出现模式中没有的字符时就可以将模式、正文大幅度地"滑过"一

收稿日期:2006-0-1;修回日期:2006-01-0

段距离.

设 ∑为固定字符集,BM 算法的关键是,对给定的模式 P,定义一个从字母到正整数的映射(函数)

$dist: c \to \{1,2,3,\cdots,m\}$,这里 $c \in \sum$

函数 dist 称为滑动距离函数,它给出正文中可能出现的任意字符 c 在模式中的位置.

主要思想[8]:假设将正文中自位置 i 起"往左"的一个子串与模式进行自右至左的匹配比较过程中,若发现不匹配(不管在何位置),则下次应从正文的 $i + dist[T_i]$ 位置重新进行 BM 算法的匹配比较工作,其效果相当于把模式、正文向右滑过一段距离 $dist[T_i]$,即跳过 $dist[T_i]$ 个字符而无需比较. 显然,若字符 T_i 不出现在模式中或者仅仅在模式末端出现,则向右滑过最大的一段距离 m.

函数 dist 的具体定义如下:

$$dist[c] = \begin{cases} m, \text{若任意字符 } c \text{ 不出现在 } P \text{ 中或者 } c = p_j (j = m), \\ \quad \text{但 } c \neq p_j (1 \le j \le m-1) \\ m - j, \text{这里 } j = \max(j : p_j = c, 1 \le j \le m-1) \end{cases}$$

如表 1 所示,是对模式串 P = "abdcd"和正文串 T = "qpamdebcdabdcd"按 BM 算法进行匹配的过程(下画线黑体为匹配失败的地方).

表 1　BM 算法的匹配过程

正文	q	o	a	**m**	**d**	e	b	c	d	a	b	d	c	d
一	a	b	d	**c**	d									
二			a	b	d	c	**d**							
三						a	b	d	c	**d**				
四										a	b	d	c	d

2.3　将 KMP 算法与 BM 算法相结合的改进算法

BM 算法和 KMP 算法各有所长,能否充分发挥二者的优点而设计出一种更有效的串匹配算法?首先考虑将 KMP 算法中确定 next 值时所采取的自左至右扫描模式的方式改成从右到左扫描模式的方式,且同时考虑模式中第 m 个字符到第 $j+1$ 个字符和正文中已经匹配及第 j 个字符与正文不匹配这两个事实,来确定出类似于 KMP 算法中的 newnext 函数,我们称此函数为 delta,定义如下:

$$delta[j] = \min_{1<s<m} \{ s : \begin{cases} \text{当 } j > s \text{ 时 } p[j+1 \ldots m] = p[j-s+1 \ldots m-s] \\ \quad \text{且 } p[j] \neq p[j-s] \\ \text{当 } j \le s \text{ 时 } p[s+1 \ldots m] = p[1 \ldots m-s] \end{cases} \}$$

$delta[j]$ 表示当正文和模式匹配到 j 位置使得 $p[j] \neq t[k]$ 时,按照 KMP 算法思想所能够向右滑动模式的最小字符个数.

经过这样的修改后,假设将正文中自位置 i 起"往左"的子串和模式进行自右至左的匹配比较过程中,当发现不匹配(例如 $p[j] \neq t[k]$)时,则从 $i = \max\{i + dist[t[i]], i + delta[j]\}$ 位置开始重新执行自右至左的匹配比较. 其效果相当于把模式串向右滑过 KMP 算法和 BM 算法所允许的距离中的一段最大距离. 具体实现时只要把 BM 算法中的语句 $i = i + dist[t[i]]$ 改为 $i = i + \max\{dist[t[i]], delta[j]\}$ 即可.

3　对传统算法的改进

3.1　一种改进的 BM 算法[6]

当模式匹配失败时,通过正文中和模式字符串的最后一个位置对应字符的下一个字符(下面的例子中为字符'i')来决定右移的字符个数. 我们将右移的字符个数记为 N,显然 $N \ge 1$,算法的关键即为求 N 的最大值. 如果字符 i 出现在模式串中,则移动的距离通过模式串决定;否则,跳过该字符,将 i 的下一个字符与模式串的最左端 P_1 对齐,重新进行比较,依次循环,直到匹配为止. 在下面的例子中,我们跳到 i 的下一个字符 n 再重新进行比较.

例如:模式串为:search

正文串为:Substrin search 匹配过程如表 2 所示:

表 2　按照该改进后的算法的匹配过程

正文	S	u	b	s	t	**r**	i	n	g	s	e	a	**r**	**c**	h
一	s	e	a	r	c	**h**									
二								s	e	a	r	c	**h**		
三										s	e	a	r	c	h

共比较 2 + 6 次.

而按照 BM 算法,匹配过程如表 3 所示:

表 3　按照 BM 算法

正文	S	u	b	s	t	**r**	i	**n**	g	s	e	a	r	c	h
一	s	e	a	r	c	**h**									
二			s	e	a	r	c	**h**							
三									s	e	a	r	c	**h**	
四										s	e	a	r	**c**	h

共比较 3 + 6 次. 显然,移动次数有所改进.

而对于正文串:subtsrbbti 与模式串:srbbti,分别利用 BM 算法及 BM 的改进算法进行匹配,分析过程如表 4 及表 5 所示,BM 算法却又比 BM 的改进算法要快.

表 4　按 BM 算法

正文	s	u	b	t	s	**r**	b	b	t	i
一	s	r	b	b	t	i				
二					s	r	b	b	t	i

表 5　按 BM 改进后的算法

正文	s	u	b	t	s	**r**	**b**	b	**t**	i
一	s	r	b	b	t	**i**				
二				s	r	b	b	t	**i**	
三					s	r	b	b	t	i

由以上分析不难看出,对 BM 算法的上述改进与 BM 算法本身各有所长. 所以可以将 BM 算法与上述 BM 改进算法进行结合,取二者的最大值来进行移动. 该算法的设计思想如下.

3.2　BM 算法的进一步改进

根据上面的分析不难看出,如果将 BM 算法与 BM 改进后(算法 3.1)的算法相结合,当不匹配发生时,取其移动距离的最大值,匹配速度显然要比 BM 算法和 BM 改进后的算法都要快,且可以避免两种算法的各自不足. 但是如果我们直接将两种算法的移动距离进行比较,显然又要在比较上花费额外的

时间,这样使结合后的算法效率未必会比单个算法的效率有所提高.

为了解决这个矛盾,可以设计一个如下的二维数组,并对其进行初始化,对于任意字符 x、y 属于集合 $\sum$,有

$$a[x][y]=\begin{cases}\text{dist}[x], & \text{当 dist}[y]+1<\text{dist}[x]\\ \text{dist}[y]+1, & \text{当 dist}[x]\leq\text{dist}[y]+1\end{cases}$$

其中 $\text{dist}[x(y)]=$

$$\begin{cases}m, & \text{若任意字符 } x(y) \text{ 不出现在 } P \text{ 中或者 } x(y)=p_j(j=m),\\ & \text{但 } x(y)\neq p,(1\leq j\leq m-1)\\ m-j, & \text{这里 } j=\max\{j:p_j=x(y),1\leq j\leq m-1\}\end{cases}$$

对于任意字符 x、y 属于集合 $\sum$,该二维数组中的每一项记录着字符序列 xy 对应的最大移动距离.假设将正文串自位置 i 起"往左"与模式串进行自右至左的匹配过程中,若发现不匹配(不管在何位置),按照本改进后的算法,则下次应从正文的 $i+a[T_i][T_{i+1}]$ 位置重新进行匹配比较工作.

它用字符值作为数组的下标,数组的大小依赖于可能出现的字符的多少,与模式串无关.如果进行中文关键字的匹配,需要扩充到整个 ASCII 码字符集,数组大小则为 256 × 256.而不需要进行中文关键字匹配时,由于 ASCII 码中 0～127 为英文字母、数字及符号等可见字符,而 128～255 并未定义,所以可以定义数组的大小为:128 ×128.该二维数组的内容就是,BM 算法及 BM 改进算法(算法 3.1)分别计算出的实际移动距离的最大值.比如在表 4、表 5 中第一次匹配失败时,BM 算法中对应字符 r 的 $\text{dist}[r]$ 为 4,实际移动距离也是 4;而按照 BM 改进算法,对应 b 的 $\text{dist}[b]$ 为 2,实际移动距离为 3,对应的二维数组 $a[r][b]=\max\{\text{dist}[r],\text{dist}[b]+1\}=4$.按照此算法思想,对于正文串:subtsrbbti 与模式串:srbbti,匹配过程如表 6 所示:

表 6 按 BM 的进一步改进算法

正文	s	u	b	t	s	**r**	**b**	b	t	i
一	s	r	b	b	t	i				
二					s	r	b	b	t	i

而对于模式串:search 和正文串:Substringsearch,匹配过程如表 7 所示:

表 7 按 BM 的进一步改进算法

正文	S	u	b	s	t	**r**	**i**	n	g	s	e	a	r	c	h
一	s	e	a	r	c	**h**									
二							s	e	a	r	c	**h**			
三										s	e	a	r	c	h

可以看出,此改进后的算法比 BM 和 BM 改进算法的性能有所提高,具有一定的优越性.然后再将此改进后的算法与 2.3 中的算法相结合,即当正文中自位置 i 起"往左"的子串和模式串进行比较的过程中,当发现不匹配(例如:$p[j]\neq t[k]$)时,则从 $i=i+\max\{a[T_i][T_{i+1}],\text{delta}[j]\}$ 位置开始重新执行自右至左的匹配.具体实现时只要把 BM 算法中的语句 $i=i+\text{dist}[t[i]]$ 改为 $i=i+\max\{a[T_i][T_{i+1}],\text{delta}[j]\}$ 即可.

4 结束语

本文在对传统模式匹配算法(KMP 算法和 BM 算法)的分析基础之上,结合对传统模式匹配算法的现有改进,对 BM 算法做了进一步的改进.结果表明,此改进后的算法具有更高的匹配效率.

参考文献:

[1] Knuth D E, Morris J H, Pratt V R. Fast pattern matching in strings[J]. SIAM Journal on Computing, 1997, 6(1): 323 - 350.

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

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

[4] HORSPOOL RN. Practical fast searching in strings[J]. Software-Practice and Experience, 1980, 10(6): 501 - 506.

[5] Daniel M Sunday. A very fast substring search algorithm[J]. Commun. ACM, 1990, 33(8): 132 - 142.

[6] 伊静,刘培玉. 入侵检测中模式匹配算法的研究[J]. 计算机应用与软件, 2005, 22(1): 112 - 114.
Yi Jing, LiuPeiyu. The survey of the pattern matching algorithm in intrusion detection system[J]. Computer Applications and Software, 2005, 2(1): 112 - 114. (in Chinese)

[7] 苏璞睿,冯登国. 基于进程行为的异常检测模型[J]. 电子学报, 2006, 34(10): 1809 - 1811.
Su Purui, Feng Dengguo. An anomaly intrusion detection model based on nonhierarchical clustering[J]. Acta Electronica Sinica, 2006, 34(10): 1809 - 1811. (in Chinese)

[8] 刘功申,王永成,许欢庆. 基于字频的单模式匹配算法[J]. 电子学报, 2002, 30(12): 2079 - 2082.
Liu Gongshen, Wang Yongcheng, Xu Huanqing. A single pattern matching algorithm based on character frequency[J]. Acta Electronica Sinica, 2002, 30(12): 2079 - 2082. (in Chinese)

作者简介:

牟永敏 男,1961 年生于山东烟台,教授,主要研究方向为网络安全、嵌入式、软件自动生成技术.
E-mail: yongminmu@biti.edu.cn

李美贵 男,1982 年生于山东菏泽,硕士研究生,主要研究方向为网络安全、嵌入式.

梁 琦 女,1980 年生于辽宁抚顺,硕士,助教,主要研究方向为计算机应用.