一种改进的 Boyer-Moore 算法在 IDS 中的应用

王淅娜[1] 喻建鹏[2]
(解放军后勤学院研究生2队 北京 100858)[1] (海军装备部 北京 100841)[2]

摘 要 在IDS的检测引擎模块设计中,基于Pattern-Matching的误用检测算法是设计师们最常用到的一种核心技术实现途径,而IDS丢包率和误报率的高低以及检测引擎匹配速度的快慢都取决于模式匹配算法性能的好与坏。Boyer-Moore算法及其改进了的Boyer-Moore Horspool算法和Boyer-Moore HorspoolS算法是目前应用最广泛的单模式匹配算法。在分析了BM算法及各种改进算法的基础上提出了一种新的基于BM算法的改进算法。该算法利用了字符串末字符和末字符对应文本串的下一字符的唯一性,同时考虑了文本串的信息以加大匹配速率,从而更好地适应IDS对模式匹配算法高效性的要求。

关键词 入侵检测,BM算法,模式匹配,Snort,KMP算法
中图法分类号 TP301.6 **文献标识码** A

Improved Boyer-Moore Algorithm Applied in IDS

WANG Xi-na[1] YU Jian-peng[2]
(The 2th Postgraduate-team of PLA Logistics Academy, Beijing 100858, China)[1]
(Equipment Department of the Navy, Beijing 100841, China)[2]

Abstract In module design for detection engine of IDS, the misuse of detection based on the Pattern Matching algorithm is the most commonly used by designers as a means of core technology, and the loss rate of data packet, the rate of false positives of IDS and Matching speed of detection engine depend on the performance of Pattern Matching algorithm. Boyer-Moore algorithm, its improved Boyer-Moore Horspool algorithm and Boyer Moore Horspool System algorithm are the the most used-widely Pattern Matching algorithm. Based on the analysis of the BM algorithm and improved algorithm, a new improved BM algorithm is proposed in the article. The algorithm takes advantage of the end character of the string and the uniqueness of next character of corresponding text strings in it, and consider the text string information to increase Matching speed, so that can accommodate to the requirement of high-efficiency of Pattern Matching algorithm for IDS.

Keywords Intrusion detection, BM algorithm Pattern-matching, Snort, KMP algorithm

1 Snort 系统模式匹配算法分析

Snort是一种通过提取分析上传而来的数据包,并与已存攻击数据库中的数据包进行比对来检测入侵行为是否发生的入侵检测系统。在检测过程中,如发现该数据包与攻击数据库中的字符特征完全相符,则记录此数据包为入侵攻击行为包,然后根据此数据包提炼出其相应的特征并为系统编写相应的检测规则,进而形成一个规则的数据库,方便后续的数据报文按照此规则库进行匹配查询,如匹配查询成功,则将其视为入侵行为,并作下一步处理。如发现不相符,则让其顺利通过。在通常的入侵检测规则数据库里,会存有成千上万的字符数据,对于整个数据库来说,设计一种快捷高效的匹配算法是提高数据库检索效率的重要保证。在Snort的检测过程中,针对数据包的不同特征,需要对其进行特征字符串、首节点以及其他类型的匹配查找,而特征字符串的查找匹配是整个检测过程中最繁琐也是最重要的一项操作。相应地,应用于Snort系统中的字符串模式匹配算法的优劣将直接关系到整个安全检测系统检测效率的提高。如果大幅度地提升其算法速度,那么Snort系统的整体性能将会得到大大地提升。

2 Boyer-Moore 模式匹配算法

模式匹配算法是指从源文本 W 中查找目标模式文本 M 的一种过程。假设 M 为目标模式文本,其长度记为 m,首字符至末字符依次记为 $M_1, M_2, \cdots, M_m$,W 为需要检索匹配的源文本,其长度记为 w,首字符至末字符依次记为 $W_1, W_2, \cdots, W_w$,对于特定的位移量 δ,如果目标模式文本中的每一个字符与源文本相对应的字符完全匹配,则称匹配成功;如没有找到目标模式文本 M,则称匹配失败,并返回一个特定的标识。

2.1 Knuth Morris Pratt 算法

Knuth Morris Pratt(KMP)算法是一种在一个字符串中定位另一个字符串的高效算法,该算法由D. E. Knuth与V. R. Pratt和J. H. Morris两位教授共同合作设计并提出,因此又称为Knuth Morris Prat(简称KMP算法)。KMP算法的核心就是依据预先指定的模式串 $M_1, M_2, \cdots, M_m$ 定义一个包含了模式串本身局部匹配信息的 *next* 函数。其基本思想是

王淅娜(1980—),女,博士,主要研究方向为信息系统及安全;喻建鹏(1979—),硕士,主要研究方向为计算机应用。

当文本串 W 与模式串 M 在字符串的某个位置处不相一致时，不需要对文本串 M 的当前指针向前移动，而只考虑将模式串 M 的当前指针行向前移动，并且通过已匹配成功的部分，尽量保持将模式串 M 向右移动，然后开始下一轮的匹配。

2.2 Boyer-Moore 算法

Boyer-Moore 算法简称 BM 算法，是迄今为止应用最为广泛的单模匹配算法。20 个世纪 70 年代中期，Boyer 和 Moore 两位研究人员在其发表的论文中首次提出该算法。其算法的核心思想是首先对模式串 M 进行初期处理，计算坏字符函数 r 和好后缀函数两个偏移函数，然后将模式串 M 最左边的字符 M_1 与文本串 W 最左边的字符 W_1 相对齐，依次从右边开始向前执行匹配。当文本 W 字符与模式 M 字符不相符时，开始调用 Badchar 规则和 Goodsuffix 规则，计算出坏字符和好后缀两个函数值，并把两个函数值中的最大值作为模式串 M 向右移动的距离值。最后，模式串 M 在文本串 W 中向右移动后重新开始进行下一轮的匹配，如字符匹配成功则输出其值，否则，继续执行匹配。

(1)Badchar 规则

当模式串 M 中的末字符 $M[m]$ 与文本串 W 中的相应字符 $W[n]$($m<n$)相符时，则可根据由右向左的顺序执行匹配，当字符 $W[x]$ 与 $W[y]$ 不相等时，即 $W[x]$ 和 $M[y]$ 不相符时，根据坏字符规则需要进行如下讨论，我们可假设此时的字符 $W[x]$ 为 a，字符 $M[y]$ 为 b：

1)如果字符 a 不存在于模式串 M 中，则将模式串 M 一次性地向右移动整个字符串 M 的长度 m 个字符，然后再按照由右向左的顺序依次从最后一个字符开始向前逐执行匹配。

2) 如果字符 a 存在于模式串 M 中，为了让模式串 M 中最右端的字符 a 和文本串 W 中的字符 a 彼此相互对齐，可将模式串 M 相应向右移动，然后再根据从右向左的顺序执行匹配。

(2)Goodsuffix 规则

如果已执行完成功匹配的部分字符，存在于模式串 M 中还未执行过匹配查询的那部分字符中，则可将该区域部分字符与文本串 W 中相对应的字符相对齐，然后再按照从右向左的顺序执行新的匹配。

由上述分析可以看出，Boyer-Moore 算法与 Knuth Morris Pratt 算法的最根本的区别就是：当文本串 W 与模式串 M 在字符串的某一位置相互对齐后，Knuth Morris Pratt 算法是从所对齐字符的地方开始向后一位逐一进行匹配(该字符不一定是模式串 M 的第一个字符)，而 Boyer-Moore 算法则是从模式串 M 的最后一个字符(该字符一定是模式串 M 的最后一个字符)开始向前一位逐一与文本串执行匹配。

3 基于 BM 的改进算法

通过上述对 BM 算法的分析和研究，我们可以了解到，鉴于 BM 算法固有的特性，其主要有以下两个不足之处：其一，在好后缀的处理上很难理解和实现，仍旧存在重复比较，降低了匹配效率；其二，随着匹配规则的增多，多次的训练效果越发不明显。本节针对以上问题设计了一种利用字符串后一位字母的唯一性来提高最大位移出现概率的改进了的 BM 算法，从而大幅度提高了字符串规则匹配的速度。

3.1 改进的 BM 算法原理

目前对 BM 算法的改进研究，大部分的思路都是从坏字符和好后缀两方面来进行研究改进的，但由于种种因素，使得改进算法速度仍然不是很理想。本文的 BM 改进算法的主要思想是：

设文本串为 W，模式串为 M，如图 1 所示。

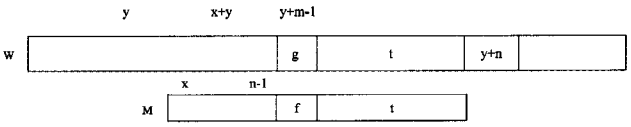


图 1 文本串与模式串框图

当 $W[x+y]\neq M[x]$ 时，即本次字符匹配查询失败后，通过查看文本串的第 $W[y+n]$ 位在模式串 M 中出现的次数再做出进一步的处理。

i)如果文本串 W 的第 $[y+n]$ 位在模式串 M 中没有出现一次，即模式串 M 中根本不存在文本串的第 $[y+n]$ 位，此时，可将模式串 M 向右移动 $n+1$ 位；

ii)如果文本串 W 的第 $[y+n]$ 位在模式串 M 中出现且仅出现一次，并且 $W[y+n-1]$ 所对应的右移量大于 $W[y+n]$ 所对应的右移量，此时，可将模式串 M 向右移动 $n+1$ 位；

iii)如果文本串的第 $W[y+n]$ 位在模式串 M 中出现的次数大于 1，此时，可通过查看 $W[y+n-1]$ 和 $W[y+n]$ 所对应的右移位数来确定模式串右移的位数。

改进了的 BM 算法流程如图 2 所示。

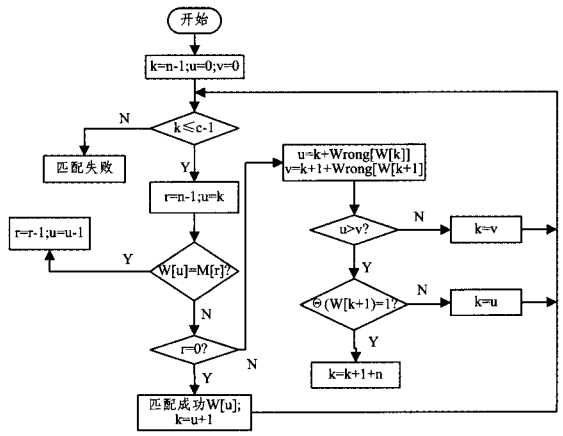


图 2 改进的 BM 算法流程图

BM 改进算法的总体思路是：对模式串的每一个字符逐位进行比对，当查找到 $M[y]$ 字符时，首先计算出 M 的最后一位字符所对应的 $W[y+n-1]$ 字符的右移位值 $y+n-1+Wrong[W[y+n-1]]$ 并将其赋于 u，然后将 $W[y+n-1]$ 字符的下一个字符的右移位值 $y+n-1+WrongW[y+n]]$ 赋于 v，最后再确定 u 和 v 的大小关系，具体分为以下两种情况：

i)如果 u 大于 v，则需要再确定 W 的第 $[y+n]$ 字符在模式串 M 中出现的次数，此时可以用 $\theta(x)$ 函数来描述 $W[y+n]$ 在模式串 M 中出现的次数。

$$\theta(x)=\begin{cases}1, & x\text{ 在 }P\text{ 中未出现或只出现一次}\\ 0, & x\text{ 在 }P\text{ 中出现多次}\end{cases}$$

1)若 $x=1$，即 $W[y+n]$ 字符在 M 中仅匹配出现一次或

根本没有出现，则可把 M 向右直接移动 $n+1$ 位，然后进行下一轮的比较；2)若 $x=0$，即 $W[y+n]$ 字符在 M 中匹配出现的次数大于 1，则将 M 的最后一位和 $W[u]$ 字符相对齐，然后开始下一轮的查询匹配；

ii)如果 u 小于等于 v，则将模式串 M 的最后一位和文本串的 $W[v]$ 字符相对齐，然后开始下一轮的查询匹配。

BM 算法和改进了的 BM 算法字符查询匹配移位情况如表 1 和表 2 所列。

表 1　算法匹配过程

W	u	g	a	i	t	c	h	w	f	m	h	t	o	u	q	f	g	h	z	c	h	c	j	h	z	f	m	a	t	c	h
M	z	f	m	a	t	c	h																								
2					z	f	m	a	t	c	h																				
3								z	f	m	a	t	c	h																	
4															z	f	m	a	t	c	h										
5																	z	f	m	a	t	c	h								
6																								z	f	m	a	t	c	h	
7																									z	f	m	a	t	c	h

表 2　改进了的 BM 算法匹配过程

W	u	g	a	i	t	c	h	w	f	m	h	t	o	u	q	f	g	h	z	c	h	c	j	h	z	f	m	a	t	c	h
M	z	f	m	a	t	c	h																								
2									z	f	m	a	t	c	h																
3																	z	f	m	a	t	c	h								
4																									z	f	m	a	t	c	h

从表 1 和表 2 两种算法的匹配移位对比可以看出，对于相同的文本串 W，BM 算法匹配进行了 7 轮才匹配成功，模式串的最大位移量为 n，且最大移位量 n 也仅仅出现了 2 次，也就是说在整个匹配的过程中，最大匹配位移出现的概率是 28.7%，即不到 29%，而改进了的 BM 算法只进行了 4 次位移查询匹配就查找到了目的字符串。在匹配过程中，不仅每次模式串的位移量是 $n+1$，而且最大位移出现的概率是 100%。很明显，与传统的 BM 算法相比，改进了的 BM 算法在最大位移量及其出现的概率上，都明显优越于传统的 BM 算法。

4　实验对比分析

4.1　测试环境

硬件平台：CPU/AMD Fusion APU E2-1800，内存/DDR32GB，硬盘/SATA320G。

软件环境：操作系统/Microsoft Windows XP Professional SP3，入侵检测系统/Snort2.8.5，网络访问系统/Winpcap 版本 4.1.3。

由于本实验是为了测试应用于 Snort 系统之中的 3 种不同算法的性能情况，即在相同的网络流量情况下分别对 KMP 算法、BM 算法和改进的 BM 算法进行测试，因此本实验把准备测试的 3 种算法以插件的形式安装在 Snort 系统中，以便测试结果更具有可比性。同时关闭 Snort 的预处理器和日志功能，使其只具有数据包匹配功能，这样检测时，Snort 就可以直接从本地硬盘上读取测试数据集，从而更好地满足系统网络流量可重复性采集的要求。

4.2　测试数据源的选取

本次实验测试所用到的数据源来自于美国麻省理工学院的林肯实验室的“DARPA1999 入侵检测数据集”。该数据集是一种基于 MITLL 采集，并由哥伦比亚大学 IDS 实验室整理而形成的安全审计数据集 KDDCUP99。该数据收集了 9 周时间的 TCPdump 网络连接和系统审计数据，仿真各种用户类型、各种不同的网络流量和攻击手段，使它更像一个真实的网络环境，这样的数据同时也解决实验的可重复性、用户隐私的保护、评测数据集公开发布等一系列问题。

4.3　实验结果分析

测试 1　我们使用经过预处理的 KDDCUP99 数据集中随机选取的 10%的标记数据作为网络样本数据，进行对混合类攻击的测试，在本测试中，每次把测试规则按 150 条进行递增，分别测量记录 3 种算法的时间和空间消耗，一直递增到 1200 条结束。为了排除某些随机因素对实验结果产生的影响，我们把每个测试进行 5 次再取其平均值。时间性能和空间性能的测试结果如图 3、图 4 所示。

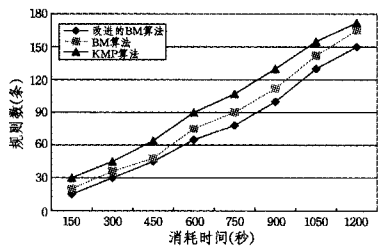


图 3　时间性能测试

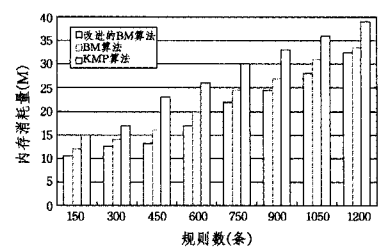


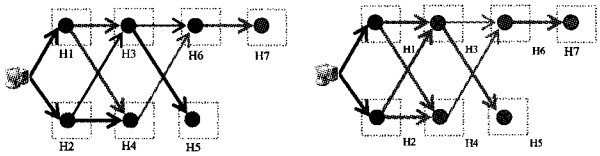
图 4　空间性能测试

通过以上测试可以看出，BM 改进算法优化了 Snort 系统，使 Snort 系统的性能得到了一定的提升，无论是在检测时间方面还是内存消耗方面都优于之前使用的 KMP 算法和 BM 算法的性能。

因为单模式匹配算法主要用在 Snort 系统对规则选项中的 content 和 uricontent 内容的检测上，并且由于 Snort 系统同时使用了多模式匹配算法，使得优化后的系统性能提升并不是很明显，但如果单独使用单模式匹配算法将会有很明显的效果。

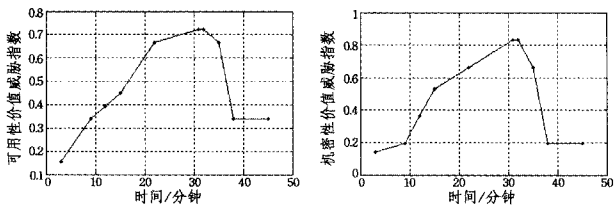
（下转第 218 页）

根据主机的脆弱性信息、入侵检测系统的攻击纪录和防御行为纪录生成攻击路径图，如图 4 所示。

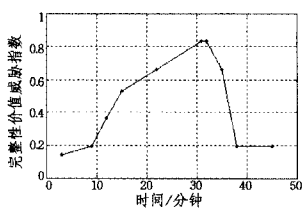


(a) 攻击路径图中的攻击行为　　(b) 攻击路径图中的防御行为

图 4



(a) 可用性价值威胁指数　　(b) 保密性价值威胁指数



(c) 完整性价值威胁指数

图 5

图 5 显示了遭受攻击后网络资产价值受到的威胁情况，当 H0 的攻击行为逐步成功后，网络的安全受到了越来越大的威胁。发现网络安全受到威胁后，通过修改防火墙 3 和防火墙 2 的访问策略，限制子网 1 到子网 2 的访问，阻断了 H0 的攻击路径，从而降低了 H0 主机对网络的威胁。

结束语　本文针对网络攻防对抗实时变化的特点，提出一种基于攻防对抗的网络安全动态评估方法。该方法对攻击过程和防御过程中网络设备的脆弱性状态变化进行量化评估，能较好地实现网络攻防过程的可视化。网络管理人员可以根据本文提供的结果直观、准确地了解当前的网络安全状态，从而为下一步采取的应对措施提供辅助决策支持。

参 考 文 献

[1] 韦勇，连一峰. 基于日志审计与性能修正算法的网络安全态势评估模型[J]. 计算机学报，2009，32(4)：763-772

[2] Lau S. The spinning cube of potential doom[J]. Communications of the ACM，2004，47(6)：25-26

[3] 徐玮晟，张保稳，李生红. 网络安全评估方法研究进展[J]. 信息安全与通信保密，2009，50(4)

[4] 马俊春，王勇军，孙继银，等. 基于攻击图的网络安全评估方法研究[J]. 计算机应用研究，2012，29(3)

[5] Yu D，Frincke D. Improving the quality of alerts and predicting intruder's next goal with hidden colored Petri-net[J]. Computer Networks，2007，51(3)：632-654

[6] Aven T. A unified framework for risk and vulnerability analysis covering both safety and security[J]. Reliability Engineering and System Safety，2007，92(6)：745-754

[7] 廖年冬，易禹，胡琦. 动态实时网络安全风险评估研究[J]. 计算机工程与应用，2011，47(36)

[8] 陈锋，刘德辉，张怡，等. 基于威胁传播模型的层次化网络安全评估方法[J]. 计算机研究与发展，2011，48(6)：945-954

(上接第 198 页)

测试 2　在 KDDCUP99 入侵检测数据集中根据测试需要对网络流量进行分割，使得分割后的测试数据文件中只包含某一种特定的攻击。本测试从 4 大类典型攻击中随机选取以下 8 种特定攻击：Dos 类选取 pingofdeath 和 smurf；R2L 类选取 imap 和 named；U2R 类选取 overflow 和 Perl；PROBING 类选取 Nmap 和 IPsweep。

测试仍然采用进行 5 次再取其平均值的方式，测试结果如图 5 所示。

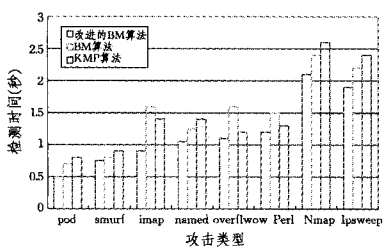


图 5　特定攻击检测

从本组测试结果可以看出：对于 imap 攻击、overflow 攻击和 Perl 攻击而言，KMP 算法的检测速度比 BM 算法分别快了 12.5%，24.8%，13.3%。而对于其它几种攻击，BM 算法的检测速度都比 KMP 算法快了近 7.6%～12.5%。而改进的 BM 算法对每一种攻击的时间性能都明显要优于其它两种算法。

结束语　Snort 的成功是建立在对源代码不断改进完善的基础之上的，因此本章提出了一种对 BM 的改进算法，增大了每轮匹配的移位量，并使用麻省理工学院林肯实验室的 KDD99 数据集作为数据源，对 3 种算法进行了离线的测试，针对混合类攻击和特定类攻击两方面的测试表明，采用本文的 BM 改进算法之后，Snort 的时间性能和空间性能都得到了优化，效率得到了一定的提升。

参 考 文 献

[1] 冉占军，姚全珠. 模式匹配算法在入侵检测中的应用[J]. 计算机应用技术，2011，21(12)：63-65

[2] 王新志，等. 一种面向软件行为可信性的入侵检测方法[J]. 中国科学技术大学学报，2011，41(7)：626-635

[3] 李雪莹，刘宝旭，许榕生. 字符串匹配技术研究[J]. 计算机工程，2011，30(22)：24-26

[4] 杨文君，魏占国，王玉平. 入侵检测系统中高效的模式匹配算法[J]. 小型微型计算机系统，2010，30(11)，2281-225

[5] Namjoshi K，Narlikar G. Robust and Fast Pattern Matching for Intrusion Detection [C]// IEEE Conference Computer Communications. Piscataway，2010：14-19

[6] Kim H J，Hong H，Kim H-S，et al. A Memory-Efficient Parallel String Matching for Inrusion Detection System [J]. IEEE Comunications Letters，2010，13(12)：1004-1006

[7] 郇正军. 基于 snort 的网络入侵检测系统研究[D]. 山东大学，2009：18-19

[8] 袁静波，郑吉森，丁顺利. 一种 BM 模式匹配算法的改进[J]. 计算机工程与应，2009，45(17)：105-107