The Improvement of MWM Multiple Pattern Matching Algorithm

Qiang Zheng
School of Computer Science & Technology，Shandong University of Technology， Zibo China 255049
Email: our.net@163.com

Abstract: This paper firstly introduces some famous multiple pattern matching algorithms and puts emphasis on the basic idea and the implementation principle of the Wu-Manber algorithm. An improvement to the Wu-Manber algorithm is provided to solve the shortcomings of Wu-Manber algorithm. Experimental result shows this algorithm can effectively expedite the speed of pattern matching compared to the traditional methods， mi proves the efficiency of the intrusion detection.

Keywords: intrusion detection ； Wu-Manber algorithm ； pattern matching；string matching

MWM 多模式匹配算法的改进

郑 强
山东理工大学计算机学院，淄博，中国，255049
Email: our.net@163.com

摘 要: 本文首先介绍了一些著名的多模式匹配算法，重点介绍了 Wu-Manber 算法的基本概念及其实现原理。并针对 Wu-Manber 算法存在的不足，提出了一种改进的 Wu-Manber 算法. 实验表明，该方法与传统模式匹配方法相比能有效地加快模式匹配的速度，提高入侵检测效率。

关键词: 入侵检测；模式匹配；Wu-Manber 算法；字符串匹配

引言

多模式串匹配算法是网络内容分析中的一个关键算法。随着网络带宽的不断扩大，网络流量不断增加，迫切需要效率更高、更适于实时处理的算法，因此研究串匹配算法是具有重要意义的。典型的 NIDS 是用一系列的规则(特征)来描述已知的攻击行为。据统计，现在大约 95%的入侵检测都是特征匹配的入侵检测。由此可见，模式匹配算法性能的好坏直接影响到入侵检测系统的效率。

本文主要对目前常用的多模式匹配算法 Wu-Manber 算法进行了分析，提出了一种更好的改进算法，显著提高了模式匹配的效率。

1．入侵检测中的模式匹配算法

模式匹配：是指在给定长度为 n 的目标串 $T = T_1T_2...T_n$ 中查找长度为 m 的模式串 $P= P_1P_2...Pm$ 的首次出现或多次出现的过程。这里 T_i $(1\le I \le n)$，$P_j(1 \le J \le m)\in\sum$(字符集)，若 P 在 T 中出现 1 次或多次，则称匹配成功，否则称匹配失败。单模式匹配算法：在目标串中 1 次只能对 1 个模式串进行匹配的算法。 多模式匹配算法：在目标串中可同时对多个模式串进行匹配的算法。

1.1 Boyer-Moor 算法

Boyer-Moor 算法[1]被认为是日常应用中效率最高的模式匹配算法，BM 算法分为预处理和搜索 2 个阶段，预处理阶段需要 o（m+σ）的时间和空间，搜索阶段需要 o（mn）的时间。

在预处理阶段，算法利用好后缀和坏字符 2 个移动规则计算匹配窗口的移动距离。在搜索阶段，匹配窗口从左向右移动，在窗口内的扫描从右向左进行，并利用当前匹配窗口中的已匹配字符以及匹配失败的字符，查找预处理好的好后缀移动表和坏字符移动表，使匹配窗口向右移动较大的距离，跳跃过文本串中不需要对比的字符. 这两种移动方法的效率都很高，例如快速搜索算

法只使用坏字符移动就可获得不错的效果。在使用坏字符移动时有可能出现负的移动，即窗口会向左移动这是不允许出现的。一般情况下我们会向右移动一位，这是安全的但是效率比较低。

在 BM 算法中通过取好前缀与坏字符两种移动的最大值解决了这一问题(因为好前缀移动总为正)，但这样做等于忽略了坏字符移动没有从本质上解决问题。

1.2 Quick-Search 算法

QS(Quick-Search)算法,是BM算法的一个变种,只采用了 BM 算法中的不良字符转移机制。它是基于下列观察,当遇到不匹配的字符时,模式总是向右移动至少一个字符,但从不超过 m 个字符。这样,在下一次匹配时总是要检测字符 Text[s+m],因此,可以将不良字符转移机制应用于 Text[s+m],来获得更大的跳跃距离。

1.3 Wu-Manber 算法

Wu-Manber 算法[2]是 BM 算法在多模式匹配中的派生形式，是一种快速实用的多模式串匹配算法。在多模式串匹配中，随着模式串数量的增多，各个字符出现在模式串后端的概率也相应增大，与串尾的距离逐渐缩小，模式串的移动距离也减小，所以 BM 算法的效率在多模式串匹配中被大大地削弱。Wu-Manber 算法采用字符块 B 扩展坏字符移动规则来解决这个问题，同时用散列表来筛选匹配阶段应进行匹配运算的模式，减少算法匹配时间，提高运行效率。

Wu-Manber 算法首先对模式串集合进行预处理．预处理阶段将建立 3 个表格：SHIFT 表，HASH 表和 PREFIX 表．SHIFT 表存储着字符集中所有字符块在模式串中出现时的移动距离．计算移动距离时有以下 2 个原则：

1)如果窗口中的字符块 WB 不出现在任何模式串中，则 SHIFT [h]=m-B+1，其中，h 为字符块 W_B 的 hash 值。

2)如果 W_B 出现在某些模式串中，而且在所有的模式串中的最右出现位置为 q，则 SHIFT[h]=m-q。HASH 表用来存储尾块字符散列值相同的模式串，PREFIX 表用来存储尾块字符散列值相同的模式串的首块字符散列值。匹配阶段就是利用这 3 个表来完成文本的扫描和寻找匹配的过程。

算法匹配的主要过程：

1)计算所有模式串中最短的模式串的长度，记为 m，并且只考虑每一个模式串的前 m 个字符，即 m 为匹配窗口的大小；

2)根据文本当前正考察的 m 个字符，计算其尾块字符 T[m-B+1，□ ，m]的散列值 h；

3)检查 SHIFT [h]的值，如果 SHIFT [h]>0，将窗口向右移动 SHIFT [h]大小位置，返回第(2)步，否则，进入第(4)步；

4)计算文本中对应窗口“前缀”的散列值，记为 text_ prefix；

5)对符合 HASH[h]≤P<HASH[h+1]的每一个 P 值，检验是否存在 PREFIX[P]=text_prefix．如果相等，对文本和模式串进行完全匹配．

Wu-Manber 算法的时间复杂度在平均情况下是 O(Bn / m)。其中，B 是字符块的长度，n 是文本串的长度，m 是模式的最短长度．该算法对最短模式长度比较敏感，SHIFT 表的最大值受模式串最小长度的限制，如果最小长度很小，则移位的值不可能很大，对匹配效率有很大影响。

文献[3]在 snort2. 6. 0 中实现了一个 Wu-Manber 算法的变型 ModifiedWu-Manber 算法(简称 MWM)。MWM 使用了一个标准的 1 或 2 字节坏字符跳跃表，一个固定的 2 字节前缀 Hash 表。值得指出的是，基于 Boyer-Moore 跳跃思想的各种算法，最大优点就是字符比较可以跳跃进行，但是其性能对最小模式长度有很高的依赖性。因为模式的最大跳跃距离不能超过该值，否则就会漏掉可能的匹配。最小模式长度越长，坏字符跳跃越有效，查找速度也就越快。

分析 snort 规则集发现，许多规则组的大部分规则是多字节模式(本文中指长度大于等于 3 字节的模式，下同)，但是也包含有 1 或 2 字节模式的规则。这些 1 或 2 字节模式尽管数量少，却导致整个模式组的最大跳跃距离为 1，即没有跳跃，从而大大降低了 Wu-Manber 算法的性能。本文提出了一种新的 Wu-Manber 类型的多模式匹配算法，我们称之为 QWM(QuickWu-Manber)。QWM 把模式分组，对 1 或 2 字节的模式采用位图方式匹配，使得剩下模式组的最小模式长度大于 2，从而显著地提高了算法性能。

2 改进的 Wu-Manber 算法的设计

提高 Wu-Manber 算法效率的主要途径是增大匹配窗口的移动距离，减少字符对比次数。改进的 QWM 算法把模式分组，对 1 或 2 字节的模式采用位图方式匹配，使得剩下模式组的最小模式长度大于 2，从而显著地提高了算法性能。

978-1-935068-18-1 © 2010 SciRes.

由于模式匹配算法在检查每一个数据包时，只要确定当前字符串与模式组中的某个模式匹配，则报告该数据包可能含有攻击，不再检查该数据包中剩余字符，而立即开始检查下一个数据包。与长模式相比，短模式被匹配的可能更大一些，所以 QWM 算法首先用短模式组与数据包匹配，再用长模式组与数据包匹配。一旦发现某个数据包中包含与模式匹配的字符串，则直接转入对下一个数据包的检查。

对于含有 1 或 2 字节模式的模式组，首先把模式组划分为两个子模式组:短模式组 $P_{shorter}$= { P_1， P_2}和长模式组 P_{longer}= {P_m， P_{m+1}，…， $P_{k|m}$ >2}。在读入规则文件时，把 1 字节模式、2 字节模式和多字节模式分别添加到相应的模式结构 OneBytePat、TwoBytePat 和 MultiBytePat 中。对于不含 1 或 2 字节的模式组，则不划分该模式组，并且只构造 MultiBytePat。本文只讨论含有 1 或 2 字节模式的模式组。

2.1 预处理阶段：

首先对短模式组 $P_{shorter}$ 进行预处理。对每个出现在 P1 中的字符 char，在大小为 256 的存在位图 EXISTONE 中把相应位置标记为 1。对每个出现在 P2 中的字符对 char1char2，在大小为 256×256 的存在位图 EXISTTWO 中把相应位置标记为 1。这样，在查找阶段就可以直接根据存在位图中相应位置的值是否为 1 来迅速确定文本中的当前字符是否与 P_1 或 P_2 中的某个模式匹配。

EXISTONE 和 EXISTTWO 可以按照如下方法构造:

$$\text{EXISTONE(char)} = \begin{cases} 1 & char \in p_1 \\ 0 & char \notin p_1 \end{cases}, \qquad (1)$$

$$\text{EXISTTWO(char1， char2)} = \begin{cases} 1 & char1char2 \in p_2 \\ 0 & char1char2 \notin p_2 \end{cases} \qquad (2)$$

2.2 模式与文本匹配的算法

2.2.1 短模式组中的模式与文本匹配的算法过程

1) 以文本中当前字符 tc 为索引，检查 EXISTONE(tc)。

2)以文本中当前 2 个字符 tc， tc+1 为索引，检查 EXISTTWO(tc， tc+1)。

3)如果 EXISTONE(tc)和 EXISTTWO(tc， tc+1)的值都为 0，文本指针加 1，转第 1 步。

4)如果当前模式区分大小写，检查在区分大小写情况下是否匹配。如果不匹配，文本指针加 1，转第 1 步。

5)报告匹配，结束查找。

2.2.2 长模式组中的模式与文本匹配的算法过程

对于长模式组 P_{longer}，首先对模式排序，然后为其构造一个坏字符跳跃表 BCSHIFT 和一个前缀哈希表 Hash。

1) 以文本中当前字符 tc 为索引，查找 BCSHIFT(tc)。

2)如果 BCSHIFT(tc)的值大于 0，文本指针右移 BCSHIFT(tc)个字符，转第 1 步。

3)计算文本中当前 2 个字符 tc， tc+1 的 Hash 函数值，以 Hash(tc)为索引查找 Hash 表，若其值为 0，文本指针加 1，转第 1 步。

4)检查与当前模式有相同前缀的所有模式。如果没有模式与当前文本匹配，文本指针加 1，转第 1 步。

5)如果当前模式区分大小写，检查在区分大小写情况下是否匹配。如果不匹配，文本指针加 1，转第 1 步。

6)报告匹配，结束查找。

QWM 算法把短模式从模式组中分离出来先作处理，长模式组的最小模式长度大于 2，也即最大跳跃距离至少为 3，从而显著地缩短了每个数据包的检查时间，提高了模式匹配算法的性能。

3 仿真实验

为了检验新算法的性能，将 QWM 算法和 snort 中实现的 MWM 算法进行了实验对比。实验环境采用 Pentium4 的 PC 计算机作为硬件平台，CPU 为 2. 0GHz，内存为 256MB; 操作系统: red hatlinux 9. 0，在 snort2. 6. 0 版本下进行实验。测试数据采用麻省理工林肯实验室开发的 DARPA 1999 入侵检测数据集[4]。该数据集覆盖了 Probe、DoS、R2L、U2R 和 Data 等五大类 58 种典型攻击方式，是目前最为全面的攻击测试数据集。实验结果中的所有时间数据都是通过 Linux 下的“time”命令测量 10 次取平均值得到的。

图 1 中所用的规则集都包含长度为 1 和 2 的模式，从图 1 可以看出， QWM 算法的执行时间明显低于 MWM 算法，算法性能明显提高。当实验采用规则集是包含长度为 1 和 2 的模式的 1000 条规则， 在检测不同的真实流量时， FWM 算法的性能均比 MWM 算法有不同程度的提高，但是当最小模式长度大于 2 时，两个算法的性能差距很小。

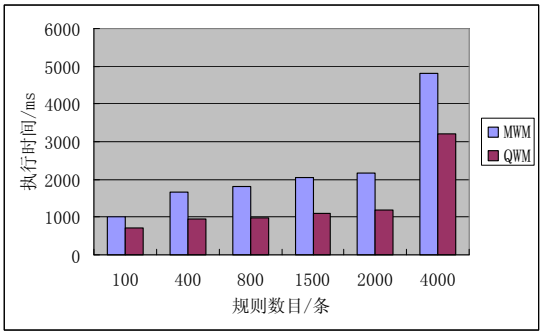

图 1 规则数目不同时性能分析

4 结束语

本文对 ModifiedWu-Manber 算法进行了改进，把模式分组， 对不同子模式组采用不同的匹配方法，对 1 或 2 字节的模式采用位图方式匹配， 对长模式使用改进的 QuickWu-Manber 算法，减少了模式集与文本的匹配次数，提高了匹配效率.

致 谢

在这里首先对在我写作过程中给予帮助的所有人表示感谢.没有他们的鼓励和无私的支持，我将不可能完成该论文。

另外由衷感谢我的家人，是他们在我的生命旅程中一直关心、帮助和支持我。

References (参考文献)

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

[2] WU S，MANBER U．A fast algorithm for multi-pattern searching Tech Rep TR94—17[R]．Tucson：Department of Computer Science，University of Arizona，1994．

[3] Snort2. 6. 0 [EB/OL]. [2006-12-05]. http: //www. snort. org/d.l

[4] 1999 DARPA intrusion detection evaluation data set[DB /OL].[2007-04-09]. http: //www. l.l mit. edu/IST/ideval/ data/1999 /1999_data_index. htm.

978-1-935068-18-1 © 2010 SciRes.